

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**OPTIMALIZACE OPERAČNÍHO SYSTÉMU S JÁDREM
LINUX PRO ZAŘÍZENÍ NAPÁJENÉ Z AKUMULÁTORU**

LINUX KERNEL OPERATING SYSTEM OPTIMIZATION FOR BATTERY-POWERED DEVICES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Přemysl Lefler

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Benedikt

BRNO 2021

Diplomová práce

magisterský navazující studijní program **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Přemysl Lefler

ID: 195379

Ročník: 2

Akademický rok: 2020/21

NÁZEV TÉMATU:

Optimalizace operačního systému s jádrem Linux pro zařízení napájené z akumulátoru

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s jádrem operačního systému Linux a prostudujte dostupné distribuce pro procesorovou architekturu ARM. Prostudujte možnosti optimalizace operačního systému pro snížení spotřeby zařízení, na kterém bude operační systém spuštěn. Operační systém by měl po zásahu vykazovat funkčnost jako senzor, nebo jednotka pro vyhodnocování dat ze senzorové sítě. V rámci optimalizace spotřeby je možné využít i externí řízení uspávání systému pomocí mikrokontroleru. Celou problematiku řešte na jednodeskovém počítači Raspberry Pi, kde bude snížení spotřeby ověřeno měřeními. Jako referenci spotřeby je brán původní operační systém bez uživatelského zásahu.

Výstupem práce budou výsledky ve formě dat a grafů o prokazatelném snížení spotřeby, včetně postupů, jak snížení spotřeby docílit. Dále jako výstup práce definujte testovací scénář, kterým ověříte snížení spotřeby.

DOPORUČENÁ LITERATURA:

[1] CULIC, Ioana, Alexandru RADOVICI a Cristian RUSU. Commercial and Industrial Internet of Things Applications with the Raspberry Pi. Berkeley, CA: APress, 2020. ISBN 978-1-4842-5296-3.

[2] WARD, Brian. How Linux Works, 2nd Edition: What Every Superuser Should Know. 2. San Francisco, CA: No Starch Press, 2014. ISBN 978-1-59327-567-9.

Termín zadání: 1.2.2021

Termín odevzdání: 24.5.2021

Vedoucí práce: Ing. Jan Benedikt

prof. Ing. Jiří Mišurec, CSc.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá optimalizací operačního systému s jádrem Linux pro akumulátorově napájené zařízení Raspberry Pi 4. Raspberry Pi 4 nabízí oproti jiným zařízením, běžně používaných v rámci Internet of Things projektů, mnoho funkcí v rámci jednoho zařízení. Nevýhodou je však velká spotřeba elektrické energie. Cílem diplomové práce je dosáhnout co největší úspory ve spotřebě elektrické energie zařízení Raspberry Pi 4, s ohledem na funkčnost zařízení jako serveru pro sběr dat z čidel.

KLÍČOVÁ SLOVA

Linux, Raspberry Pi 4, optimalizace, spotřeba

ABSTRACT

This thesis focuses on optimization of Linux-based operating system for a accumulator-powered device Raspberry Pi 4. Compared to other devices commonly used in Internet of Things projects, the Raspberry Pi 4 offers many functions within one device. However, the disadvantage is the high consumption of electricity. The aim of this thesis is to achieve greatest possible savings in electricity consumption of the Raspberry Pi 4 device, with regard to functionality of the device as a server for data collection from sensors.

KEYWORDS

Linux, Raspberry Pi 4, optimization, power consumption

LEFLER, Přemysl. *Optimalizace operačního systému s jádrem Linux pro zařízení napájené z akumulátoru*. Brno, 2021, 61 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Jan Benedikt

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Optimalizace operačního systému s jádrem Linux pro zařízení napájené z akumulátoru“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Janu Benediktovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Obsah

Úvod	11
1 Raspberry Pi	13
1.1 Raspberry Pi 4 model B	13
2 Vybrané Linuxové distribuce	14
2.1 Raspberry Pi OS	14
2.2 Ubuntu	14
2.3 CentOS	15
3 Měřicí přístroj UM24C	16
4 Softwarové nástroje	17
4.1 The Stress–Terminal User Interface (s-tui)	17
4.2 Top	17
4.3 PowerTOP	18
4.4 Eventstat	18
4.5 Perf	18
4.6 Iotop	18
5 Měření a optimalizace spotřeby elektrické energie	19
5.1 Porovnání spotřeby Linuxových distribucí ve výchozím nastavení . . .	19
5.1.1 Výsledky měření spotřeby zařízení ve výchozím nastavení . . .	19
5.2 Porovnání spotřeby distribucí po provedení základních úprav pro sní- žení spotřeby	23
5.2.1 Kontrola zatížení zařízení	23
5.2.2 Popis provedení základních úprav pro snížení spotřeby	24
5.2.3 Výsledky měření spotřeby zařízení po provedení základních úprav pro snížení spotřeby zařízení	27
5.3 Porovnání spotřeby distribucí po docílení minimální činnosti zařízení	30
5.3.1 Kontrola a optimalizace aktivních procesů a zápisů do inter- ního úložiště zařízení	30
5.3.2 Výsledky měření spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště zařízení	34
5.4 Zhodnocení docílených výsledků optimalizace a výběr nejúspornější Linuxové distribuce	36
5.4.1 Porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení optimalizace	37

6	Externí řízení napájení	40
6.1	Witty Pi 3	40
7	Simulace nasazení zařízení jako server pro koncentraci dat z různých čidel	44
7.1	Zpracování naměřených dat z čidel	46
7.2	Měření spotřeby zařízení v rámci testovacího scénáře	47
8	Akumulátorové napájení	49
9	Ovlivnění funkčnosti zařízení optimalizačními úpravami	50
10	Shrnutí dosažených výsledků	51
	Závěr	52
	Literatura	54
	Seznam symbolů, veličin a zkratk	57
	Seznam příloh	59
A	Obsah elektronické přílohy	60

Seznam obrázků

1.1	Raspberry Pi 4 model B.	13
3.1	Ukázka měřicího zařízení UM24C.	16
4.1	Ukázka uživatelského rozhraní aplikace s-tui.	17
5.1	Graf spotřeby při nečinnosti ve výchozím nastavení.	20
5.2	Graf spotřeby při přenosu souborů ve výchozím nastavení.	20
5.3	Graf spotřeby při 100% zatížení procesoru ve výchozím nastavení.	21
5.4	Graf průměrné hodnoty spotřeby při 100% zatížení procesoru ve výchozím nastavení.	22
5.5	Aplikace s-tui ukazující zatížení procesoru (ze shora zleva Raspberry Pi OS, Ubuntu, CentOS).	23
5.6	Graf spotřeby při nečinnosti po provedení základních úprav.	27
5.7	Graf spotřeby při přenosu souborů po provedení základních úprav.	28
5.8	Graf spotřeby při 100% zatížení procesoru po provedení základních úprav.	28
5.9	Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při nečinnosti zařízení.	34
5.10	Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při přenosu souborů	35
5.11	Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při 100% zatížení procesoru zařízení.	35
5.12	Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář s minimální zátěží.	38
5.13	Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář se střední zátěží.	38
5.14	Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář s maximální zátěží.	39
6.1	Modul Witty Pi 3.	41
6.2	Graf efektivity přenosu elektrické energie v závislosti na napájecím napětí udávána výrobce modulu Witty Pi 3 (převzato z [26]).	42
7.1	Grafické znázornění stavů zařízení Raspberry Pi v závislosti na čase.	45
7.2	Graf teploty ve fiktivním objektu <i>chata</i>	46
7.3	Graf porovnání spotřeby zařízení v rámci jednoho cyklu.	47
7.4	Graf porovnání spotřeby zařízení během 24 hodin.	48

Seznam tabulek

5.1	Přehled mediánových hodnot spotřeby jednotlivých distribucí ve výchozím nastavení.	22
5.2	Výsledné hodnoty testu pomocí nástroje <i>7-Zip</i>	27
5.3	Přehled mediánových hodnot spotřeby jednotlivých Linuxových distribucí po provedení základních úprav pro snížení spotřeby.	29
5.4	Tabulka naměřených hodnot po provedení optimalizace aktivních procesů a zápisů do interního úložiště zařízení.	36
5.5	Tabulka porovnání hodnot po provedení změn v jednotlivých distribucích vedoucích ke snížení spotřeby zařízení Raspberry Pi 4 model B.	37
5.6	Tabulka porovnání referenčních a finálních hodnot spotřeby pro Linuxovou distribuci Raspberry PI OS.	39
6.1	Tabulka možných nastavení s použitím symbolu <i>?</i> bez použití nastavení vteřin.	43
7.1	Porovnání spotřeby zařízení v rámci testovacího scénáře ve výchozím stavu a po provedení optimalizace.	48
8.1	Doba provozu zařízení při napájení z akumulátoru.	49

Seznam výpisů

5.1	Vypnutí nepotřebných rozhraní.	24
5.2	Vypnutí audio a Bluetooth rozhraní.	24
5.3	Vypnutí LED diod.	24
5.4	Přenastavení frekvence procesoru.	25
5.5	Přenastavení procesorového guvernéra pro distribuci CentOS.	25
5.6	Doporučení nástroje <i>PowerTOP</i>	25
5.7	Úpravy doporučené nástrojem <i>PowerTOP</i>	25
5.8	Ukázka příkazu pro spuštění aplikace <i>eventstat</i>	30
5.9	Příkaz pro zastavení tvorby záznamů o systémových událostech.	31
5.10	Příkaz pro zastavení procesu auditd u distribuce CentOS.	32
5.11	Příkaz pro odstranění procesů <i>avahi</i> a <i>thd</i> u distribuce Raspberry Pi OS.	32
5.12	Příkaz pro zastavení nástrojů <i>packagekitd</i> a <i>snapd</i> u distribuce Ubuntu.	32
5.13	Příkaz pro zastavení nástrojů <i>tuned</i> a <i>chronyd</i> u distribuce CentOS.	32
5.14	Příkaz pro spuštění nástroje <i>Perf</i>	32
5.15	Výstup nástroje <i>Perf</i>	33
5.16	Výstup nástroje <i>fdisk</i>	33
5.17	Výstup nástroje <i>Iotop</i>	34
6.1	Ukázka terminálového uživatelského rozhraní Witty Pi 3.	43

Úvod

Tato diplomová práce se zabývá optimalizací, porovnáním a výběrem nejúspornějšího operačního systému s Linuxovým jádrem pro zařízení Raspberry Pi 4 model B tak, aby bylo zařízení možné nasadit jako server pro sběr dat z různých čidel.

V teoretické části práce je popsáno zařízení Raspberry Pi, vybrané distribuce s Linuxovým jádrem a softwarové nástroje použité k docílení minimální spotřeby elektrické energie zařízení. Vybranými distribucemi jsou Raspberry Pi OS (Operating System), Ubuntu a Centos. Distribuce Raspberry Pi OS byla vybrána vzhledem ke skutečnosti, že se jedná o oficiální distribuci pro zařízení Raspberry Pi. Distribuce Ubuntu byla vybrána vzhledem ke skutečnosti, že stejně, jako Raspberry Pi OS vychází z distribuce Debian. Distribuce CentOS byla vybrána zejména proto, že na rozdíl od obou předešlých distribucí vychází z distribuce Red Hat Enterprise Linux. Tato část práce je dále doplněna o popis měřicího zařízení UM24C, které bylo použito pro všechny měření spotřeby zařízení.

V praktické části práce je provedeno referenční měření spotřeby elektrické energie zařízení pro všechny vybrané distribuce, v jejich výchozím nastavení, následně jsou popsány základní úpravy vedoucí ke snížení spotřeby elektrické energie a realizovány další měření spotřeby. V této části práce je také uskutečněno měření dopadu těchto optimalizačních úprav na výpočetní výkon zařízení. V další části práce jsou popsány a realizovány úpravy aktivních procesů a zápisů do interního úložiště spolu s měřením spotřeby elektrické energie po uskutečnění všech úprav. Každé měření probíhalo pro 3 různé scénáře. V prvním scénáři je zařízení v nečinnosti, jedná se tedy o scénář s minimální zátěží. Ve druhém scénáři probíhá přenos souborů do zařízení skrze SFTP (Secure File Transfer Protocol) protokol, čímž je docíleno střední zátěže zařízení. V posledním scénáři dochází ke 100% zatížení procesoru pomocí softwarového nástroje a jedná se tedy o scénář s maximálním zatížením zařízení. Po provedení měření jsou výsledná data v každé části srovnána. Po uskutečnění všech optimalizačních úprav a všech měření spotřeby elektrické energie je stanovena nejúspornější distribuce. Tato nejúspornější distribuce je dále použita v rámci simulace nasazení zařízení Raspberry Pi jako serveru pro sběr dat z různých čidel.

V práci je dále popsáno řešení externího napájení pomocí zařízení Witty Pi 3, díky kterému je dosaženo dalšího snížení spotřeby elektrické energie zařízení. V poslední praktické části práce je navržen a realizován scénář pro simulaci reálného nasazení zařízení jako serveru pro sběr dat z různých čidel. Simulace je doplněna o měření spotřeby zařízení ve výchozím nastavení a po realizaci optimalizačních úprav. Z naměřených hodnot je dále stanovena kapacita baterie pro nepřetržitý provoz zařízení, po dobu alespoň jednoho měsíce. Následně je spočteno, o kolik je prodloužena životnost zařízení při použití optimalizačních úprav oproti použití zařízení ve výchozím nastavení.

V práci jsou dále popsány všechny negativní dopady optimalizačních úprav na funkčnost zařízení spolu se shrnutím dosažených výsledků úspory spotřeby elektrické energie.

1 Raspberry Pi

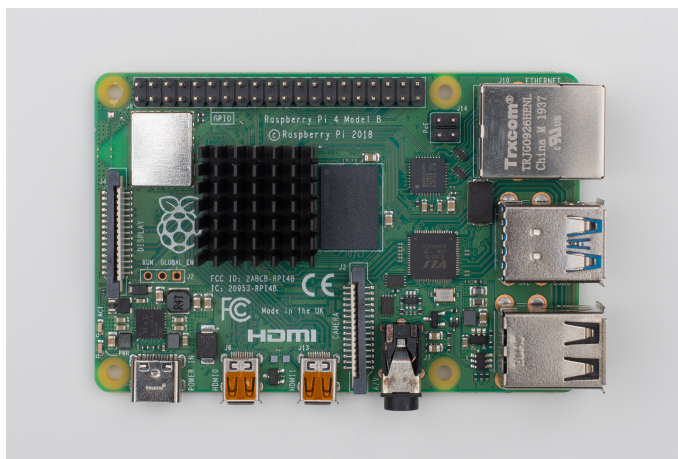
Raspberry Pi jsou malé, jednodeskové počítače, vyvinuté britskou společností Raspberry Pi Foundation. Prvotním cílem bylo přiblížit výuku informatiky ve školách a v rozvojových zemích. Časem se ukázalo, že zařízení má mnohem větší potenciál a Raspberry Pi se začalo používat i v robotice, monitoringu počasí a ve spoustě dalších odvětvích. V dnešní době se jedná o jeden z nejvíce prodávaných britských počítačů. Všechny verze tohoto zařízení jsou vybaveny procesory postavenými na architektuře ARM [1].

I když byly první modely Raspberry Pi z většiny předkonfigurovány jako bezhlavé počítače (předpokládalo se, že nebude připojen monitor, klávesnice ani myš), mohou k zařízení být připojeny USB (Universal Serial Bus) zařízení (úložiště, klávesnice, myši, atd.), monitory skrze HDMI (High-Definition Multimedia Interface) rozhraní a spoustu dalších periferních zařízení přes další rozhraní a piny, které jsou na desce.

1.1 Raspberry Pi 4 model B

Raspberry Pi 4 je v době psaní práce nejnovější verzí z řady počítačů Raspberry Pi. Tato verze byla na trh uvedena v červnu 2019. Kromě zvýšení výkonu a kapacity operační paměti (různé verze až s 8 GB) přináší podporu 4K videa, napájení skrze rozhraní USB-C a podporu USB 3.0.

Praktická část práce je prováděna na modelu Raspberry Pi 4 model B - 4GB RAM a s hliníkovým chladičem (14×14×10 mm) na procesoru.



Obr. 1.1: Raspberry Pi 4 model B.

2 Vybrané Linuxové distribuce

Zamýšlené nasazení Raspberry Pi s upraveným operačním systémem pro co nejmenší spotřebu, je akumulátorově napájený server pro sběr dat z různých čidel. Kvůli těmto požadavkům byly jako kandidáti vybrány Linuxové distribuce Raspberry Pi OS, Ubuntu, CentOS.

2.1 Raspberry Pi OS

Raspberry Pi OS (v minulosti označován jako Raspbian), je oficiální operační systém pro Raspberry Pi, který je distribuován zdarma. Raspberry Pi OS je komunitně vyvíjen. Mezi jeho hlavní výhody patří dobrá optimalizace pro ARM procesory. Jedná se o operační systém s monolitickým jádrem. Základní verze je nainstalována s mnoha výukovými programy (například Wolfram Mathematica). Poslední verze distribuce vychází z Linuxové distribuce Debian (Buster). Pro instalaci DEB (Debian) balíčků se používá APT (Advanced Package Tool) manažer [2].

V práci je použita verze August 2020 LITE bez grafického rozhraní a s minimálním počtem předinstalovaných aplikací.

Tato distribuce byla zvolena, protože se jedná o oficiální operační systém pro zařízení Raspberry Pi OS.

2.2 Ubuntu

Jedná se o jednu z nejznámější a nejpoužívanějších domácích i serverových Linuxových distribucí, která je vydávána zcela zdarma. Díky tomu je s ním spojena rozsáhlá a aktivní komunita. Mezi největší výhody patří i pravidelně vycházející aktualizace (což může mít ale i nežádoucí efekt v podobě snížení stability systému). Ubuntu je založeno na distribuci Debian. Stejně, jako Raspberry Pi OS, má monolitické jádro. Ubuntu je distribuce vyvíjená společností Canonical a komunitou dalších vývojářů. Pro instalaci DEB balíčků se zde opět používá APT balíčkový manažer [3].

V rámci Ubuntu se software rozděluje na čtyři sekce (komponenty). Jejich rozdělení je následující:

- Main – obsahuje pouze balíčky, které vyhovují licenčním podmínkám a jejich podpora je zabezpečená týmem Ubuntu (všeobecně všechny balíčky potřebné pro správný běh systému).

- Restricted – v této komponentě jsou balíčky podporované týmem Ubuntu (jsou důležité pro chod systému), ale nejsou dostupné pod vhodnou licenci a proto nemohou být zahrnuté v Main komponentě (binární balíčky grafických karet a ovladačů).
- Universe – software, který není podporován Ubuntu týmem. Může či nemusí mít restriktivní licenci. Velké množství balíčků, které si uživatelé mohou instalovat, ale zároveň jsou oddělené od komponenty Main.
- Multiverse – balíčky, které nejsou podporované a nesplňují požadavky na svobodný software [4].

Od verze 14.04 Trusty Tahr - LTS (Long-Term Support) začíná podpora pro ARM procesory. V rámci práce byla použita verze server 20.04.1 Focal Fossa - LTS.

Distribuce Ubuntu byla vybrána, protože se jedná o jednu z nejrozšířenějších Linuxových distribucí a podobně, jako je tomu u Raspberry Pi OS, vychází z distribuce Debian.

2.3 CentOS

CentOS (podobně jako je tomu u Ubuntu) patří mezi nejznámější a nejpoužívanější Linuxové (převážně serverové) distribuce. Mezi největší výhody CentOS spadá vysoká stabilita. Vysoká stabilita s sebou však nese i jistou nevýhodu v podobě velmi málo častých aktualizací distribuce (na rozdíl od distribuce Ubuntu). CentOS vychází z Red Hat Enterprise Linux (RHEL) a na rozdíl od něj, je vydáván zdarma. Původně byla podporována vlastní komunitou vývojářů, ale v roce 2014 došlo ke spojení s firmou Red Hat (zůstává nezávislá na RHEL). Obdobně jako v předchozích případech, je zde použito monolitické (modulární) jádro. Pro instalaci RPM (Red Hat Package Manager) balíčků je zde využit Yum (Yellowdog Updater, Modified) balíčkový manažer [5].

Od verze 8 začíná plná podpora pro procesory s architekturou ARM. V rámci práce byla použita verze CentOS Userland 8 Minimal.

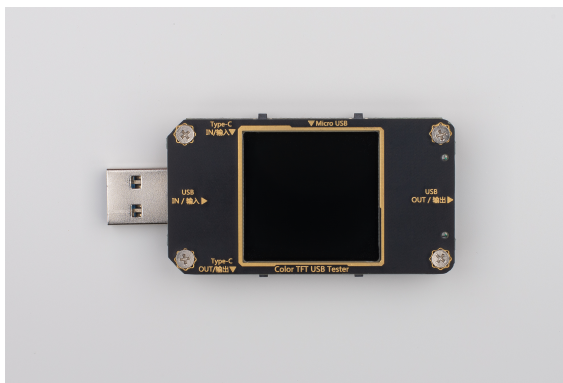
Distribuce CentOS byla zvolena, protože se jedná o nejrozšířenější serverovou distribuci, a na rozdíl od distribucí Raspberry Pi OS a Ubuntu, vychází z Linuxové distribuce Red Hat Enterprise Linux.

3 Měřicí přístroj UM24C

UM24C je přístroj pro měření napětí a proudu skrze rozhraní USB-A a USB-C, kdy do vstupu je připojené napájení a k výstupu je připojené samotné zařízení, pro které chceme dané měření provádět. Je vybaven i barevným LCD displejem, na kterém je schopen zobrazovat grafy aktuálních hodnot během měření.

Podstatné technické parametry uváděné výrobcem:

- rozsah měření napětí: 4,50–24,00 V,
- rozlišení měření napětí: 0,00 V,
- přesnost měření napětí: $\pm 0,2 \%$,
- rozsah měření proudu: 0,000–3,000 A,
- rozlišení měření proudu: 0,001 A,
- přesnost měření proudu: $\pm 0,8 \%$ [6].



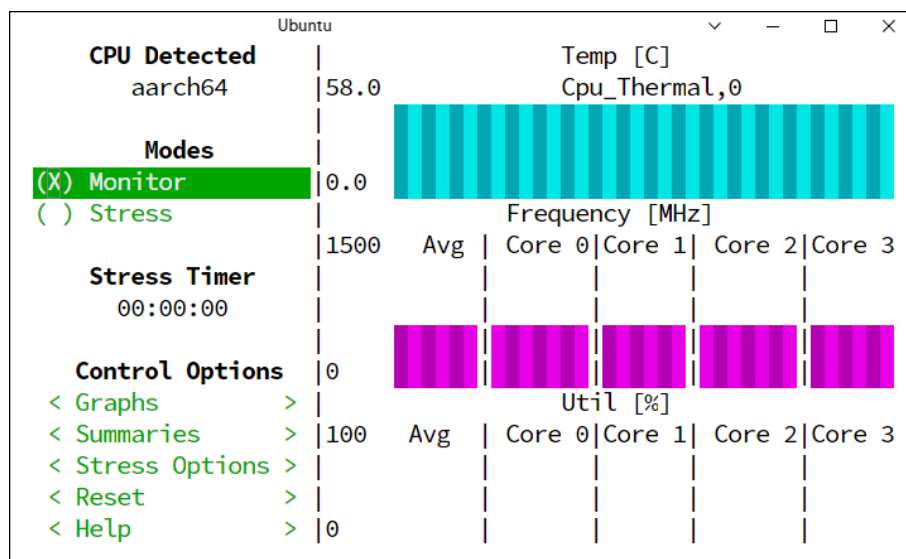
Obr. 3.1: Ukázka měřícího zařízení UM24C.

Samotné zařízení lze jednoduše připojit k počítači pomocí Bluetooth a aplikace *RD-USB*, která umožňuje data zobrazit v grafech a ukládat ve formátu CSV pro další zpracování.

4 Softwarové nástroje

4.1 The Stress–Terminal User Interface (s-tui)

S-tui je nástroj pro kontrolu zatížení, frekvence a teploty procesoru. Dále nabízí možnost pro testování stability systému formou zatížení procesoru na 100 % výkonu. Aplikaci lze spustit přímo v terminálu a nabízí i možnost jednoduchého a přehledného grafického rozhraní, jehož ukázka je na obrázku 4.1.



Obr. 4.1: Ukázka uživatelského rozhraní aplikace s-tui.

Aplikace nabízí dva stavy provozu. Prvním z nich je *Monitor* ve kterém jsou pouze zobrazovány informace o provozu procesoru (teplota, frekvence a zatížení). Druhým je *Stress*, ve kterém dochází k maximálnímu zatížení procesoru (stále zobrazuje i informace o procesoru) [7].

V rámci práce byl tento nástroj primárně používán pro 100% zatížení procesoru, pro měření scénáře se 100% zatížením procesoru.

4.2 Top

Hlavní funkcí nástroje *top* je zobrazení seznamu aktivních procesů v rámci operačního systému, který je periodicky aktualizován. Spolu s názvem a identifikačním číslem dále zobrazuje, i na kolik procent daný proces zatěžuje procesor, kolik místa zabírá v paměti, který uživatel jej spustil, nebo v jakém stavu se daný proces právě

nachází. Další užitečnou funkcí je vypnutí vybraného procesu přímo pomocí tohoto nástroje. Uživatelské rozhraní je pouze textové a celý seznam lze řadit podle požadavků uživatele [8].

4.3 PowerTOP

Softwarový nástroj sloužící pro měření a úsporu spotřeby elektrické energie počítače. Po spuštění analyzuje běžící programy, ovladače a nastavení jádra. Automaticky poté dokáže nalézt neoptimální nastavení a doporučí jejich nápravu. Tento nástroj je sice primárně určen pro laptopy, lze jej ale použít i pro zařízení Raspberry Pi [9].

4.4 Eventstat

Eventstat je užitečný nástroj, který po spuštění dokáže zachytit stav jádra systému a tím i všechny právě probíhající události. Výstupem je pak soubor ve formátu CSV, ve kterém jsou zachycené události zobrazeny.

4.5 Perf

Nástroj, který slouží pro analýzu výkonu zařízení a příčiny volání procesů jádrem. Je schopen zachytávat hardwarové i softwarové události. Využívá hardwarových registrů, díky nimž je schopen analyzovat jednotlivé instrukce systému za sebou tak, jak byly ve skutečnosti prováděny [10].

4.6 Iotop

Funkcí tohoto nástroje je monitoring využití různých vstupů a výstupů jádra operačního systému a zobrazení získaných informací do tabulky. V rámci práce byl tento nástroj použit pro monitoring zápisu dat na úložiště zařízení [11].

5 Měření a optimalizace spotřeby elektrické energie

Měření spotřeby elektrické energie probíhá v rámci této kapitoly vždy pro 3 různé scénáře, které představují různé stupně zatížení zařízení. Těmito scénáři jsou:

- Nečinnost – jedná se o scénář s minimální zátěží.
- Přenos souborů – v rámci tohoto scénáře jsou do zařízení přenášeny soubory pomocí protokolu SFTP, s omezenou rychlostí na 1 MB/s. Jedná se o scénář se střední zátěží.
- 100% zatížení procesoru – scénář s maximální zátěží, které je dosaženo použitím softwaru s-tui (toto měření není z hlediska případného nasazení zařízení tak důležité, jako dvě předešlá).

Během měření je k zařízení fyzicky připojeno vždy pouze napájení a paměťová karta s operačním systémem. Zařízení je dále připojeno k Wi-Fi síti. Mezi každým měřením je třicetiminutová časová prodleva, kdy je zařízení zcela vypnuto, aby nedocházelo k ovlivňování výsledků zvýšenou teplotou procesoru.

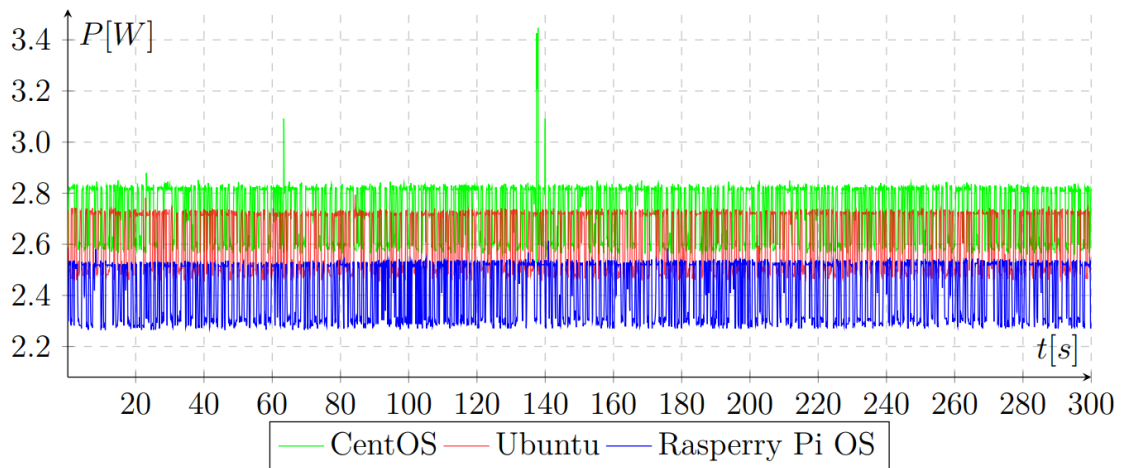
5.1 Porovnání spotřeby Linuxových distribucí ve výchozím nastavení

V této části práce je popsáno měření spotřeby jednotlivých distribucí ve stavu, v jakém jsou dodávány výrobcem.

5.1.1 Výsledky měření spotřeby zařízení ve výchozím nastavení

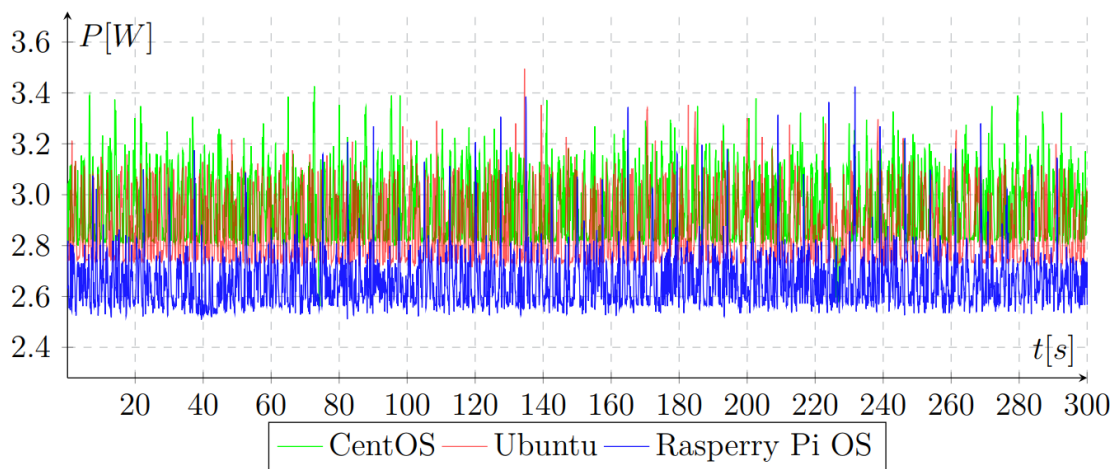
Výsledky tohoto měření můžeme vidět v tabulce 5.1 a na obrázcích 5.1, 5.2 a 5.3.

Na obrázku 5.1 si můžeme všimnout, že u distribuce CentOS došlo pro scénář s minimální zátěží celkem ke třem značným výkyvům spotřeby. Špičková hodnota těchto výkyvů dosáhla hodnoty 3,474 W. U ostatních distribucí k takovýmto výkyvům nedocházelo.



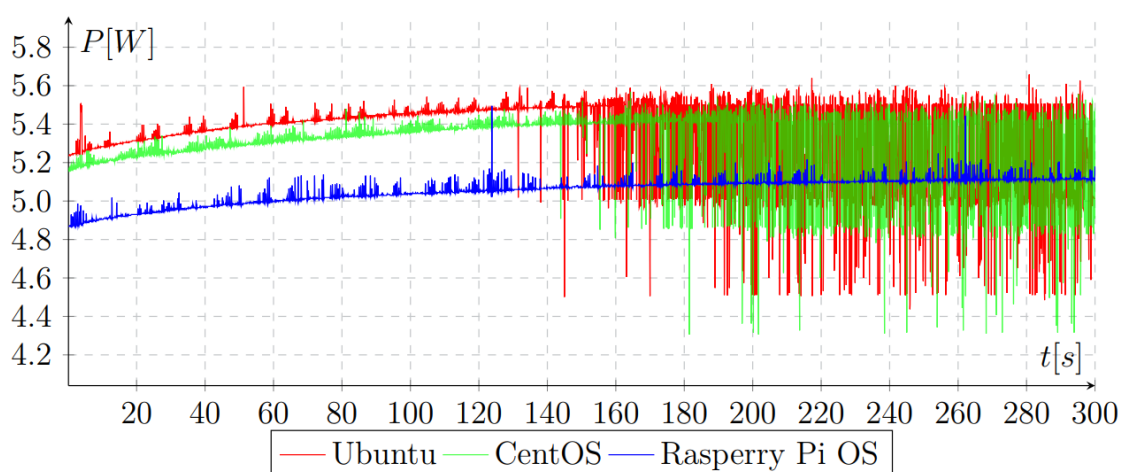
Obr. 5.1: Graf spotřeby při nečinnosti ve výchozím nastavení.

Obrázek spotřeby elektrické energie 5.2 představuje graf spotřeby pro druhý scénář, tedy scénář se středním zatížením zařízení. Vzhledem ke skutečnosti, že v rámci tohoto scénáře dochází k přenosu souborů prostřednictvím Wi-Fi rozhraní, jedná se o očekávaný průběh, kdy hodnota spotřeby elektrické energie značně kolísá.



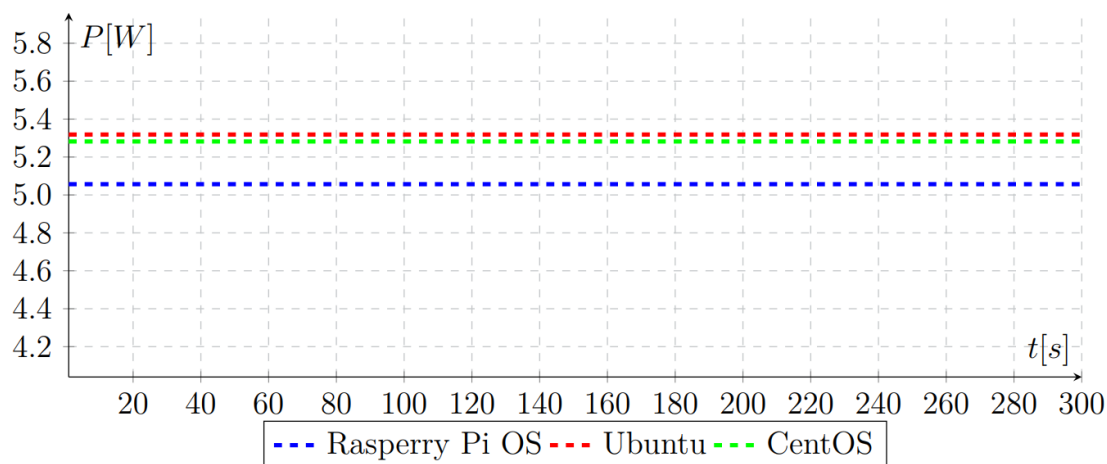
Obr. 5.2: Graf spotřeby při přenosu souborů ve výchozím nastavení.

Z obrázku 5.3 je dále patrné, že v případě distribucí Ubuntu a CentOS v druhé půlce měření došlo k poklesu výkonu procesoru a tím i ke značným výkyvům hodnoty spotřeby. Tento pokles výkonu způsobila vysoká teplota procesoru (80°C). Zařízení je nastaveno tak, že v případě dosažení kritické teploty dojde k poklesu výkonu procesoru, aby se se zařízení nepoškodilo. V případě distribuce Raspberry Pi OS nebylo v průběhu měření této teploty dosaženo, a proto nedocházelo k těmto výkyvům spotřeby. Další zajímavostí je fakt, že se spotřeba u všech distribucí postupně zvyšovala. Toto postupné zvyšování spotřeby lze přisuzovat postupně se zvyšující teplotě zařízení.



Obr. 5.3: Graf spotřeby při 100% zatížení procesoru ve výchozím nastavení.

Vzhledem k nepřehlednosti grafu na obrázku 5.3, která je způsobená poklesem výkonu procesoru zařízení je tento obrázek doplněn dalším grafem na obrázku 5.4. Na tomto obrázku jsou zobrazeny průměrné hodnoty spotřeby jednotlivých distribucí pro scénář s největší zátěží.



Obr. 5.4: Graf průměrné hodnoty spotřeby při 100% zatížení procesoru ve výchozím nastavení.

Tab. 5.1: Přehled mediánových hodnot spotřeby jednotlivých distribucí ve výchozím nastavení.

Distribuce	Stav nečinnosti	Přenos souborů	100% využití procesoru
Rasperry Pi OS	2,430 W	2,640 W	5,091 W
Ubuntu	2,714 W	2,829 W	5,407 W
CentOS	2,924 W	3,033 W	5,361 W

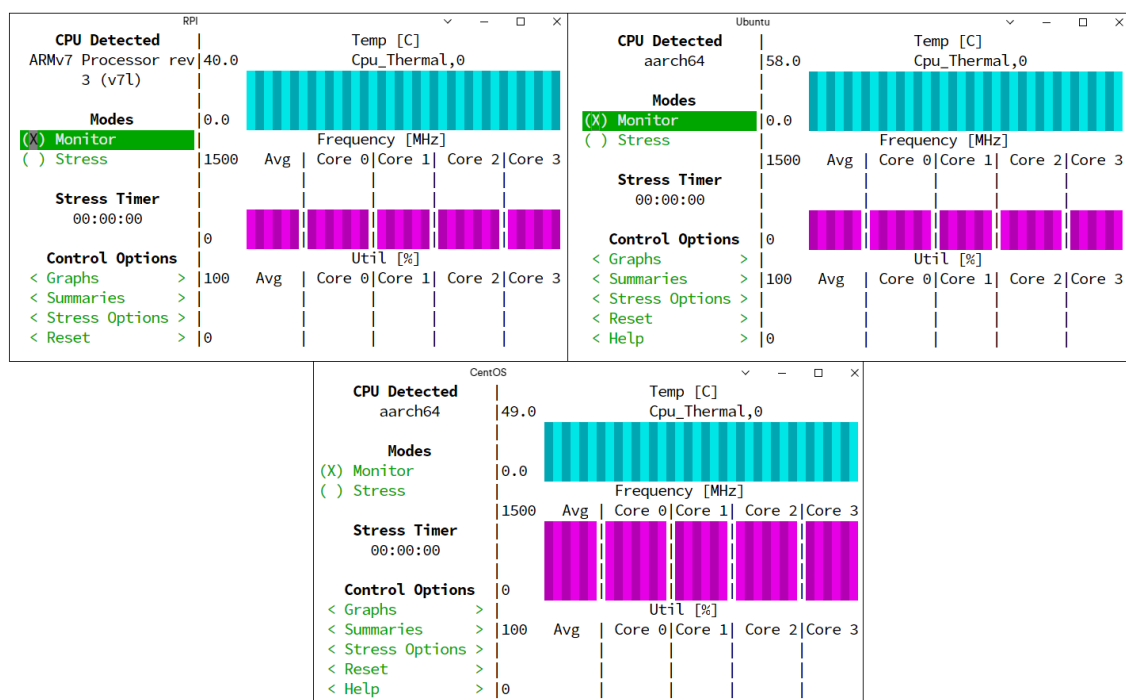
Z naměřených hodnot vyplývá, že Raspberry Pi OS je pro všechny scénáře ve výchozím nastavení nejlépe optimalizovaný. Spotřeba pro scénář s nečinností byla 2,430 W, při přenosu souborů 2,640 W a při 100% zatížení procesoru 5,091 W. Nejhorší optimalizovanou distribucí je CentOS, se spotřebou v nečinnosti 2,924 W, při přenosu souborů 3,033 W a při 100% zatížení procesoru 5,361 W. Z výsledků měření lze dále pozorovat, že při zvýšení teploty zařízení dochází i ke zvýšení celkové spotřeby. Tuto skutečnost můžeme pozorovat zejména pro scénář se 100% zatížením procesoru.

5.2 Porovnání spotřeby distribucí po provedení základních úprav pro snížení spotřeby

Každá z distribucí je předkonfigurována dle uvážení výrobce tak, aby spotřeba zařízení byla co možná nejmenší, ale zároveň nebyla nijak omezena funkčnost zařízení. I tak je zde ale možnost další úspory vypnutím nepotřebných rozhraní, snížení pracovní frekvence procesoru a dalšími úpravami, které jsou popsány v této části práce.

5.2.1 Kontrola zatížení zařízení

Při otevření aplikace *s-tui* a spuštění módu *monitor* je patrné, že žádná z distribucí nevykazovala žádnou aktivitu (obrázek 5.5). Pokud se podíváme na frekvenci procesoru, můžeme si navíc všimnout, že v případě distribuce CentOS je frekvence procesoru 1500 Mhz, zatímco u zbylých dvou distribucí je frekvence procesoru nižší (600 Mhz), což z hlediska spotřeby poukazuje na neoptimální nastavení.



Obr. 5.5: Aplikace *s-tui* ukazující zatížení procesoru (ze shora zleva Raspberry Pi OS, Ubuntu, CentOS).

5.2.2 Popis provedení základních úprav pro snížení spotřeby

Jelikož je na zařízení potřeba pouze připojení k Wi-Fi síti, lze vypnout ostatní nepotřebná rozhraní následujícími příkazy:

Výpis 5.1: Vypnutí nepotřebných rozhraní.

```
1 echo '1-1' | sudo tee /sys/bus/usb/drivers/usb/unbind
2 /opt/vc/bin/tvservice -o
3 ifconfig eth0 down
```

Příkazy uvedené ve výpisu 5.1 lze použít pro všechny vybrané distribuce. Příkazem na prvním řádku výpisu dojde k odpojení celého USB hubu v zařízení. Dalším příkazem dojde k úplnému vypnutí HDMI rozhraní a posledním příkazem se deaktivuje ethernetové rozhraní.

Audio rozhraní a Bluetooth lze vypnout přidáním dvou řádků z výpisu 5.2 do souborů */boot/config.txt* (v případě Raspberry Pi OS a CentOS) a */boot/firmware/syscfg.txt* (v případě Ubuntu).

Výpis 5.2: Vypnutí audio a Bluetooth rozhraní.

```
1 dtparam=audio=off
2 dtoverlay=disable-bt
```

Další úspory ve spotřebě je možné snadno dosáhnout vypnutím všech LED diod na zařízení. U Raspberry Pi OS a CentOS toho lze dosáhnout přidáním řádků z výpisu 5.3 do souboru */boot/config.txt* a u Ubuntu do souboru */boot/firmware/syscfg.txt*.

Výpis 5.3: Vypnutí LED diod.

```
1 dtoverlay=act-led
2
3 dtparam=act_led_trigger=none
4 dtparam=act_led_activelow=off
5
6 dtparam=pwr_led_trigger=none
7 dtparam=pwr_led_activelow=off
8
9 dtparam=eth_led0=4
10 dtparam=eth_led1=4
```

Dále je pro snížení spotřeby možné změnit maximální a minimální frekvenci procesoru. Pro všechny distribuce byla zvolena minimální frekvence 250 Mhz a maximální 600 Mhz, což by mělo být pro účel zařízení dostačující.

U všech distribucí toho lze jednoduše dosáhnout pomocí aplikace *cpupower* a zadáním příkazů z výpisu 5.4 do terminálu.

Výpis 5.4: Přenastavení frekvence procesoru.

```
1 cpupower frequency-set -u 600
2 cpupower frequency-set -d 250
```

Z obrázku 5.5 je patrné, že v případě distribuce CentOS je frekvence procesoru v klidovém režimu maximální možná. Proto je třeba přenastavit procesorového guvernéra, který je přednastaven na režim *performance*, tedy nejvyšší výkon. Přenastavení lze provést příkazem ve výpisu 5.5. V tomto nastavení bude procesor, v klidovém režimu, nastaven na minimální nastavenou frekvenci a v případě potřeby dojde ke zvýšení této frekvence (ostatní distribuce jsou takto nastaveny od výrobce).

Výpis 5.5: Přenastavení procesorového guvernéra pro distribuci CentOS.

```
1 cpupower frequency-set -g ondemand
```

Poslední úpravou v této části jsou nastavení doporučená nástrojem *PowerTOP*.

Jak je vidět z výpisu 5.6, nástroj *PowerTOP* po spuštění ihned naznačuje (u všech distribucí stejně) neoptimální nastavení zapisování dat z mezipaměti na paměťovou kartu (první řádek) a špatné nastavení sběrnice pro USB 3.0 (druhý a třetí řádek), spojené se zápisem dat do operační paměti. Protože byl v předcházejících krocích USB hub úplně vypnut, není nutné doporučení z druhého a třetího řádku věnovat pozornost. Příkaz pro provedení změny je uveden ve výpisu 5.7.

Výpis 5.6: Doporučení nástroje *PowerTOP*.

```
1 Bad  VM writeback timeout
2 Bad  Runtime PM for PCI Device Broadcom Limited Device 2711
3 Bad  Runtime PM for PCI Device VIA Technologies, Inc. VL805 USB 3.0
4 Good Wireless Power Saving for interface wlan0
5 Good Autosuspend for USB device xHCI Host Controller [usb1]
6 Good Autosuspend for USB device USB2.0 Hub [1-1]
7 Good Autosuspend for USB device xHCI Host Controller [usb2]
8 Good Wake-on-lan status for device wlan0
9 Good Wake-on-lan status for device eth0
```

Výpis 5.7: Úpravy doporučené nástrojem *PowerTOP*.

```
1 echo '1500' > '/proc/sys/vm/dirty_writeback_centisecs'
```

Po zadání tohoto příkazu dojde ke prodloužení intervalu mezi kontrolami zdali jsou v mezipaměti data vhodná k zápisu do úložiště, tedy na hodnotu 1,5 vteřiny.

Další změny, které nemají vliv na výslednou spotřebu zařízení, nebo způsobují nestabilitu systému

V době psaní diplomové práce nebyla ani na jedné z vybraných distribucí plně podporována změna napětí na jednotlivých komponentech. Při pokusu o snížení napětí na procesoru, nebo operační paměti zařízení došlo vždy ke snížení stability systému (většinou ve formě zamrznutí, nebo dokonce pádu celého systému).

Jistá úspora ve spotřebě byla očekávána při snížení pracovní frekvence operační paměti. Toto očekávání však nebylo naplněno, a při snížení pracovní frekvence operační paměti nedošlo k zaznamenanému poklesu spotřeby zařízení.

Dopad snížení pracovní frekvence procesoru na celkový výkon zařízení

V této části práce byla pracovní frekvence procesoru snížena z původní hodnoty 1,5 GHz, na maximální hodnotu 0,6 GHz a minimální hodnotu 0,250 GHz. Navíc byl v rámci všech distribucí nastaven procesorový guvernér na funkci *ondemand*. Toto nastavení znamená, že v momentě kdy není potřeba zpracovávat žádná data, procesor pracuje na frekvenci 0,250 GHz a v případě potřeby (zvýšená zátěž) je schopen zvýšit svou pracovní frekvenci až na 0,6 GHz.

Při práci se zařízením v rámci diplomové práce nebyl po provedení těchto změn pozorován žádný pokles výkonu zařízení, i když došlo k poklesu maximální pracovní frekvence o 60 %. Z tohoto pozorování lze konstatovat, že pro první scénář (tedy scénář s jen velmi malou zátěží) nedochází k ovlivnění celkového výkonu zařízení.

Při odesílání souborů skrze SFTP protokol (druhý scénář) dosahuje rychlost přenosu v obou případech své maximální rychlosti (1 MB/s), tedy ani v tomto případě není změnou pracovní frekvence provoz zařízení nijak omezen.

Jediný scénář, kde se tato změna projevuje je scénář s maximálním zatížením procesoru. K vyjádření poklesu výkonu zařízení byl použit nástroj *7-Zip*. Tento nástroj testuje výkon zařízení pomocí komprese a dekomprese souboru pomocí LZMA (Lempel-Ziv-Markov-Chain Algorithm) metody. Výstupem tohoto testu je číslo, které udává, kolik milionů instrukcí je procesor schopen provést za vteřinu (MIPS). Výsledky tohoto testu jsou zobrazeny v tabulce 5.2. Z této tabulky je patrné, že množství instrukcí, které je procesor schopen vykonat pokleslo pro provedení úprav o 59,6 %, což zhruba odpovídá očekávanému výsledku, vzhledem ke snížení maximální pracovní frekvence procesoru o 60 %.

Na základě pozorování a měření lze konstatovat, že maximální výpočetní výkon zařízení je snížen o zhruba 60 %, ale toto snížení maximálního výkonu nijak

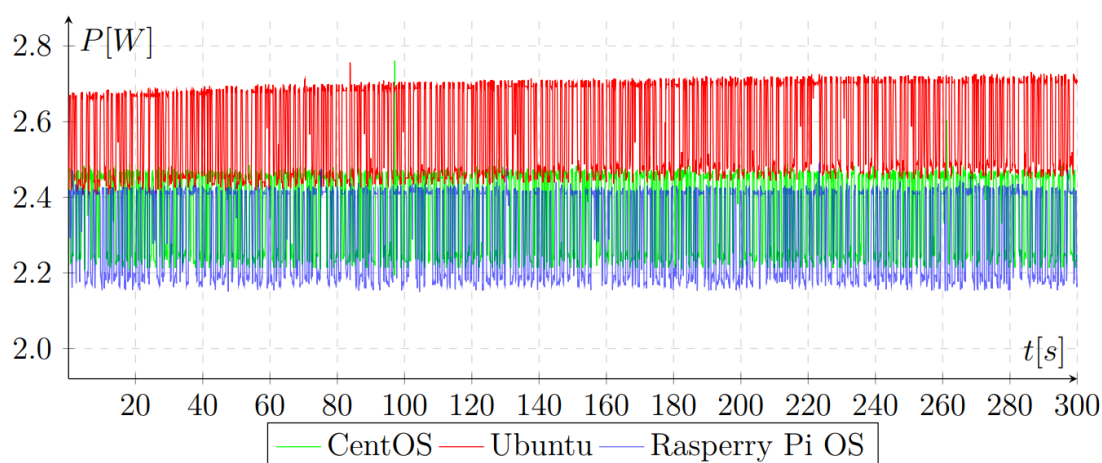
Tab. 5.2: Výsledné hodnoty testu pomocí nástroje *7-Zip*.

Maximální pracovní frekvence procesoru	MIPS
1,5 GHz	7822
0,6 GHz	3162

neovlivňuje zařízení v rámci prvního a druhého scénáře, kde nedochází k žádným změnám.

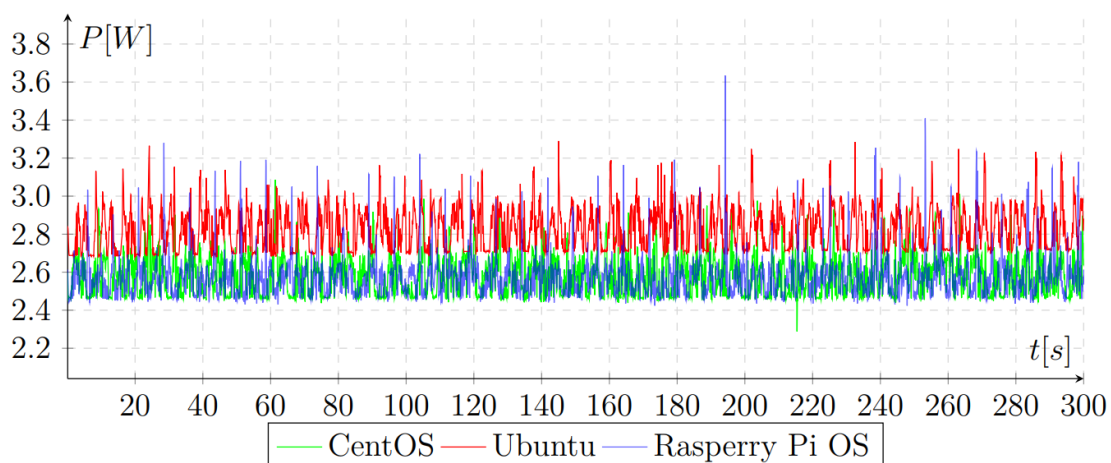
5.2.3 Výsledky měření spotřeby zařízení po provedení základních úprav pro snížení spotřeby zařízení

Výsledky měření po provedení výše uvedených úprav jsou v tabulce 5.3 a na obrázcích 5.6, 5.7 a 5.8.



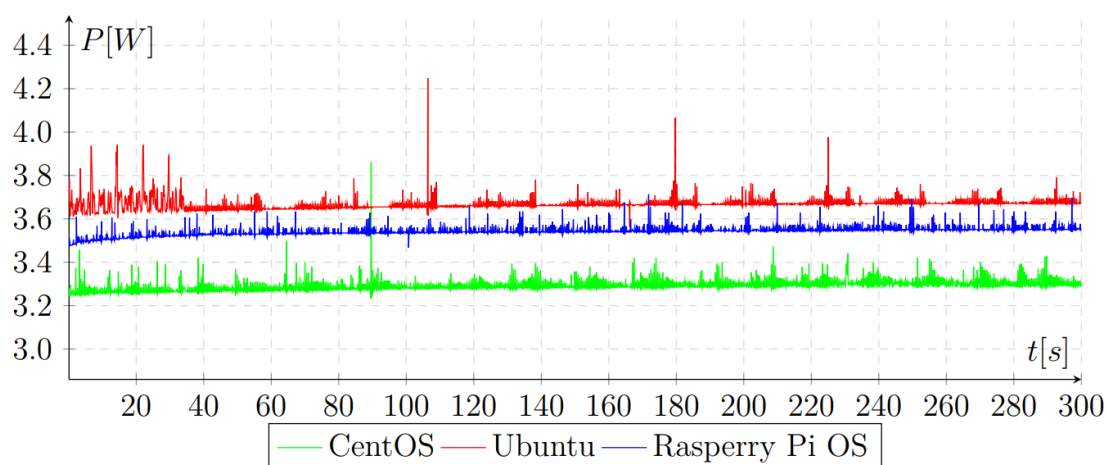
Obr. 5.6: Graf spotřeby při nečinnosti po provedení základních úprav.

Z obrázku 5.6 je patrné, že u distribuce CentOS stále dochází k velkým výkyvům spotřeby v klidovém režimu. Na rozdíl od měření z minulé části práce je patrné, že nejméně úspornou distribucí je nyní pro tento, i ostatní scénáře distribuce Ubuntu.



Obr. 5.7: Graf spotřeby při přenosu souborů po provedení základních úprav.

Z obrázku 5.8 vyplývá, že po provedení výše zmíněných optimalizačních úprav, nadále pro tento scénář nedochází k utnutí výkonu. Zařízení tedy v průběhu měření nedosahuje kritické teploty ani pro jednu z distribucí, což je zapříčiněno zejména omezením maximální pracovní frekvence procesoru.



Obr. 5.8: Graf spotřeby při 100% zatížení procesoru po provedení základních úprav.

Z výsledných grafů a tabulky 5.3 je zřejmé, že po provedení základních úprav pro snížení spotřeby došlo ke značné úspoře u všech distribucí, pro všechny scénáře. Nejvyšší úspora je u distribuce CentOS. Velkou změnou, oproti měření v předchozí části práce, je fakt, že distribuce Ubuntu je nyní pro všechny scénáře nejméně úsporná. Raspberry Pi OS je nejúspornější pro scénáře, kdy je zařízení v nečinnosti (spotřeba 2,314 W - úspora 0,116 W) a kdy probíhá přenos souborů do zařízení (spotřeba 2,557 W - úspora 0,093 W). Distribuce CentOS je nejúspornější pro scénář se 100% zatížením procesoru (3,290 W - úspora 2,081 W).

Tab. 5.3: Přehled mediánových hodnot spotřeby jednotlivých Linuxových distribucí po provedení základních úprav pro snížení spotřeby.

Distribuce	Stav nečinnosti	Přenos souborů	100% využití procesoru
Raspberry Pi OS	2,314 W	2,557 W	3,545 W
Ubuntu	2,604 W	2,819 W	3,666 W
CentOS	2,451 W	2,561 W	3,290 W

5.3 Porovnání spotřeby distribucí po docílení minimální činnosti zařízení

Tato část práce je zaměřena na optimalizaci a případné zastavení nepotřebných procesů a omezení zápisu nadbytečných dat do úložiště zařízení.

5.3.1 Kontrola a optimalizace aktivních procesů a zápisů do interního úložiště zařízení

Prvním krokem pro kontrolu všech běžících procesů v rámci operačního systému je spuštění nástroje *eventstat*. Příkaz pro spuštění toho nástroje je ve výpisu 5.8, kdy po zadání dojde k deseti výpisům všech aktivních systémových událostí do souboru *results.csv*. Jednotlivé zápisy jsou prováděny vždy po deseti vteřinách.

Výpis 5.8: Ukázka příkazu pro spuštění aplikace *eventstat*.

```
1 eventstat 10 10 -r results.csv
```

Systémové události, respektive procesy, které jsou za tyto události zodpovědné a které se objevily ve výsledném souboru jsou následující:

Procesy, které se objevovaly u všech distribucí jsou:

- **kworker** - proces zodpovědný za všechny operace na úrovni jádra (zpracování systémových volání) [12],
- **ksoftirqd** - vlákno jádra systému, které se stará (v případě, že je procesor zrovna zaneprázdněn) o řazení softwarových přerušení do fronty pro pozdější zpracování procesorem [13],
- **sshd** - proces v rámci OpenSSH serveru, který se stará o autentizaci, šifrování, terminálové připojení, přenos souborů a tunelování do zařízení a z něj pomocí protokolu SSH [14],
- **irqbalance** - proces, jehož úkolem je rozdělování hardwarových přerušení mezi jednotlivé procesory pro jejich zpracování [15],
- **systemd** - soubor nástrojů pro správu Linuxových operačních systémů. Jedná se o první proces, který je spuštěn během zavádění operačního systému (kořenový proces) [16],
- **systemd-udevd** - součást procesu *sysmtd*, tento proces zpracovává jednotlivé události v jádře systému a pro každou událost provede předem definovanou instrukci [17],
- **systemd-journal** - služba pro zapisování a následné uchovávání systémových událostí [18],

- **wpa_supplicant** - implementace IEEE 802.11i suplikantu, pro zabezpečení připojení k Wi-Fi síti,
- **cron** - proces pro automatizované spouštění procesů (skriptů).

Procesy, které se objevovaly pouze u distribuce Raspberry Pi OS jsou:

- **avahi** - proces pro komunikaci zařízení připojených k lokální síti, bez potřeby jakékoli předešlé konfigurace [19],
- **thd** - Triggerhappy je proces, který sleduje různé vstupy (stisknutí tlačítka, klávesy, přepínače, ...) a na základě těchto vstupů vykonává předem definované úkony [20].

Procesy, které se objevovaly pouze u distribuce Ubuntu jsou:

- **packagekitd** - soubor nástrojů pro snadnou instalaci a aktualizace aplikací v rámci operačního systému [21],
- **snapd** - proces pro správu balíčků v rámci operačního systému [22].

Procesy, které se objevovaly pouze u distribuce CentOS jsou:

- **tuned** - proces pro monitorování a optimalizaci nastavení v rámci operačního systému [23],
- **chronyd** - proces pro automatické nastavení lokálního času systému [24],
- **auditd** - proces pro zápis systémových událostí (jakou jsou systémová volání a přístupy k souborům) do úložiště [25].

Procesy, které jsou buď nezbytné pro správný běh operačního systému, nebo u nich není předpokládáno snížení spotřeby případnou rekonfigurací jsou *ksoftirqd*, *irqbalance*, *wpa_supplicant*, *cron*, *systemd* a *systemd-udevd*. Proto se jimi tato práce nebude dále zabývat.

Proces *sshd* zajišťuje případné připojení k zařízení pro jeho další správu pomocí protokolu *SSH*. V případě potřeby je možné aby byl tento proces zastaven, nebo odstraněn. Pro nasazení zařízení se však počítá s případnou další správou, takže je ponechán aktivní.

Naprostou většinu zápisů do úložiště zařízení tvoří logování, tedy zápis systémových událostí. Zastavení tvorby těchto záznamů lze u všech distribucí provést příkazem ve výpisu 5.9, čímž dojde i k zastavení procesu *systemd-journal*.

Výpis 5.9: Příkaz pro zastavení tvorby záznamů o systémových událostech.

```
1 systemctl disable rsyslog
```

U distribuce CentOS provádí zápis událostí i proces *auditd*, který lze zastavit příkazem ve výpisu 5.10 .

Výpis 5.10: Příkaz pro zastavení procesu auditd u distribuce CentOS.

```
1 systemctl disable auditd
```

U distribuce Raspberry Pi OS jsou navíc aktivní i procesy *avahi* a *thd*. Tyto procesy nejsou pro správné fungování zařízení nezbytně nutné, proto došlo k jejich odstranění příkazy ve výpisu 5.11.

Výpis 5.11: Příkaz pro odstranění procesů *avahi* a *thd* u distribuce Raspberry Pi OS.

```
1 apt-get autoremove avahi-daemon
2 apt-get autoremove triggerhappy
```

U distribuce Ubuntu vykazují aktivitu i procesy *packagekitd* a *snapd*. Jedná se o procesy pro správu (instalaci, aktualizace) balíčků. Proto není optimální nástroje zcela odstranit, ale pouze je zastavit do doby, než budou potřeba. Příkazy pro jejich zastavení jsou ve výpisu 5.12.

Výpis 5.12: Příkaz pro zastavení nástrojů *packagekitd* a *snapd* u distribuce Ubuntu.

```
1 systemctl disable packagekitd.service
2 systemctl disable snapd.service
```

U distribuce CentOS se dále objevovaly procesy *tuned* a *chronyd*. Protože bylo v předchozí části práce provedeno manuální nastavení spojené s optimalizací zařízení, proces *tuned* nyní neplní téměř žádnou funkci. Proces *chronyd* slouží pouze k synchronizaci času, ke správnému běhu zařízení však není nutný. tyto služby lze zastavit pomocí příkazů ve výpisu 5.13.

Výpis 5.13: Příkaz pro zastavení nástrojů *tuned* a *chronyd* u distribuce CentOS.

```
1 systemctl disable tuned
2 systemctl disable chronyd
```

Po provedení výše zmíněných změn lze nyní předpokládat, že zařízení je nyní v klidovém režimu zatěžováno pouze procesy, které jsou nezbytně nutné pro správný běh operačních systémů. Pro kontrolu je využít nástroj *Perf*. Příkaz pro spuštění tohoto nástroje je ve výpisu 5.14.

Výpis 5.14: Příkaz pro spuštění nástroje *Perf*.

```
1 perf record -e 'workqueue:*' -ag -T
```

Část výstupu nástroje *Perf* (ve skutečnosti se jednalo o několik set řádků) pro distribuci Raspberry Pi OS (u všech distribucí je výstup obdobný) můžeme vidět ve výpisu 5.15. V tomto výstupu si můžeme všimnout, že se opakují 3 záznamy, tedy proces *swapper*, *kworker* a *jbd2/mmcblk0p2*.

Swapper je proces, který je aktivní, pokud není aktivní žádný jiný proces. To znamená, že pokud je volán proces *swapper*, zařízení nic nezpracovává. Proces *Kworker* je volán jako odpověď na každé volání jiným procesem a daný úkon přiřazuje příslušnému jádru procesoru. Pokud jsou tedy volány pouze tyto dva procesy, lze konstatovat, že zařízení je v úplném klidu.

Výpis 5.15: Výstup nástroje Perf.

1	swapper	0/0
2	swapper	0/0
3	kworker/0:1H-mm	81/81
4	kworker/0:1H-mm	81/81
5	jbd2/mmcblk0p2	83/83
6	jbd2/mmcblk0p2	83/83
7	kworker/0:1H-mm	81/81
8	kworker/0:1H-mm	81/81
9	swapper	0/0
10	swapper	0/0
11	kworker/0:1H-mm	81/81
12	kworker/0:1H-mm	81/81
13	swapper	0/0
14	swapper	0/0
15	kworker/3:1-mm_	55/55
16	kworker/3:1-mm_	55/55

Dalším záznamem je *jbd2/mmcblk0p2*, což znamená, že došlo k zápisu dat do úložiště nástrojem *JBD2* (Journaling block device). Použitím nástroje *fdisk* můžeme zkontrolovat, že dané zařízení, na kterém k zápisu došlo, je paměťová karta v zařízení, konkrétně oddíl s označením *mmcblk0p2*.

Výpis 5.16: Výstup nástroje fdisk.

1	Device	Boot	Start	End	Sectors	Size	Id	Type
2	/dev/mmcblk0p1		8192	532479	524288	256M	c	FAT32 (LBA)
3	/dev/mmcblk0p2		532480	60579839	60047360	28.6G	83	Linux

JBD2 je nástrojem pro kontrolu a zápis informací o souborovém systému úložiště. Je spouštěn jádrem systému, takže jej nelze jednoduše zastavit. V případě, že by k jeho zastavení došlo, mohlo by navíc dojít ke značnému snížení stability

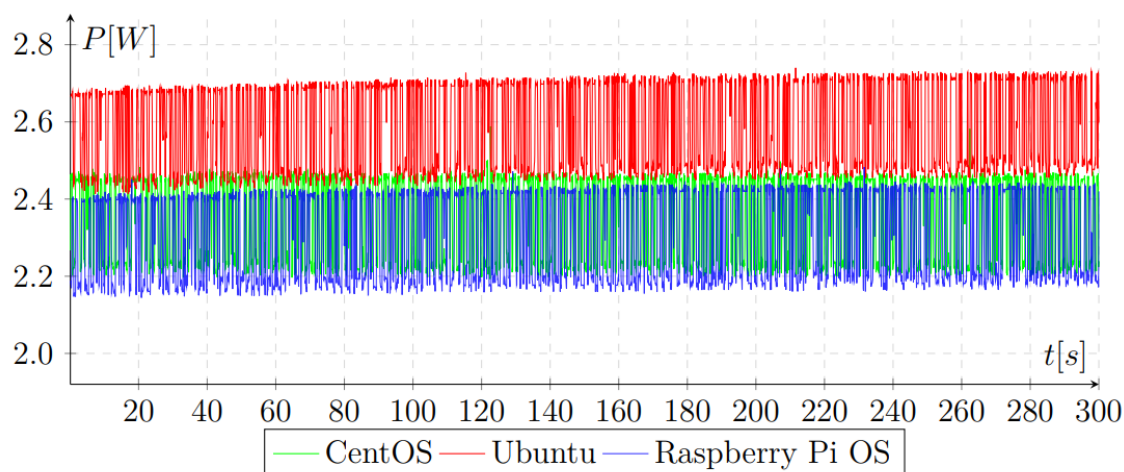
a výkonu zařízení. Pro kontrolu, jak často a jaké množství dat tento nástroj zapisuje na paměťovou kartu lze použít nástroj *Iotop*. Aplikace byla spuštěná 20 minut a jsou evidovány pouze 2 zápisy do úložiště. Z této skutečnosti lze vyvodit, že nástroj *JBD2* zařízení zatěžuje jen velmi málo a případné rekonfigurace by zřejmě znatelnou úsporu spotřeby energie nepřinesla. Výstup tohoto nástroje je ve výpisu 5.17.

Výpis 5.17: Výstup nástroje Iotop.

```
1 14:27:39 0.00 K/s 7.16 K/s 0.00 \% 0.25 \% [jbd2/mmcblk0p2-]
2 14:39:20 0.00 K/s 29.66 K/s 0.00 \% 0.40 \% [jbd2/mmcblk0p2-]
```

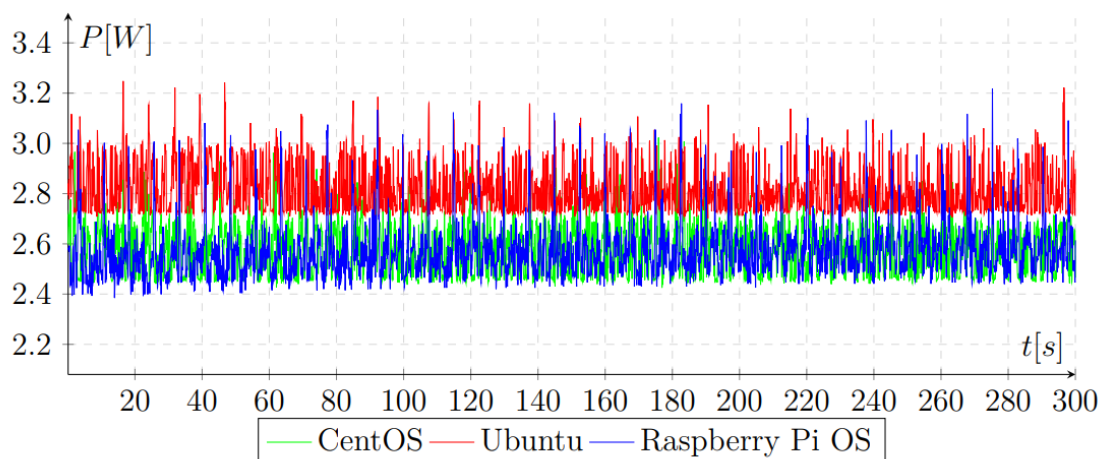
5.3.2 Výsledky měření spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště zařízení

Výsledky měření po provedení výše zmíněných úprav jsou v tabulce 5.4.1 a na obrázcích 5.9, 5.10 a 5.11.

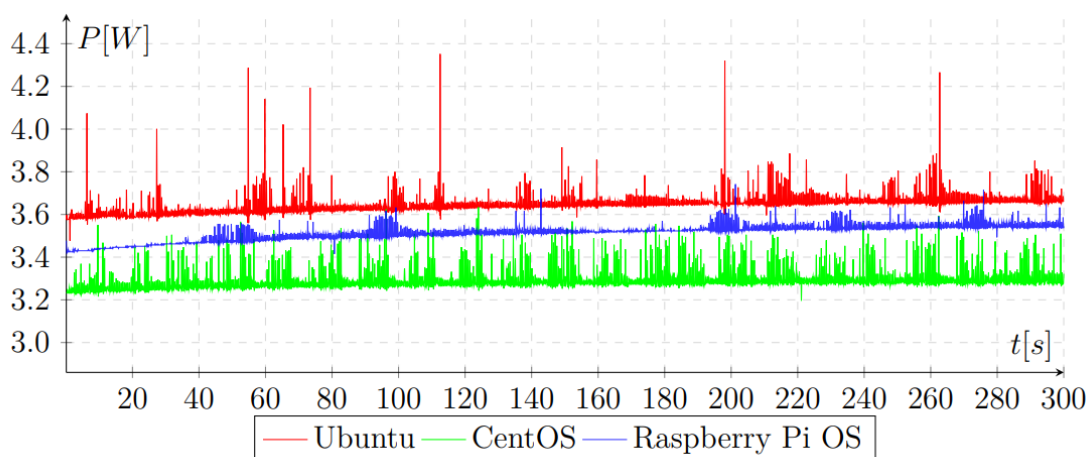


Obr. 5.9: Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při nečinnosti zařízení.

Z grafu na obrázku 5.9 je patrné, že u distribuce CentOS nadále nedochází k velkým výkyvům spotřeby, jak tomu bylo v předešlých částech práce. Tyto výkyvy byly způsobovány procesem *chronyd*.



Obr. 5.10: Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při přenosu souborů



Obr. 5.11: Graf spotřeby zařízení po provedení optimalizace aktivních procesů a zápisů do interního úložiště při 100% zatížení procesoru zařízení.

Z výsledných grafů a tabulky 5.4 je patrné, že po provedení výše zmiňovaných úprav došlo i v tomto případě k úspoře spotřeby elektrické energie, i když není tak velká, jako tomu bylo v předešlé části práce. Dále lze konstatovat, že ve všech případech je po provedení výše zmíněných úprav zařízení v klidovém režimu skutečně téměř neaktivní, až na občasné zápisy do úložiště zařízení, u kterých ale není předpokládán dopad na celkovou spotřebu zařízení podstatný. Raspberry Pi OS zůstává nejúspornější distribucí pro scénáře, kdy je zařízení ve stavu nečinnosti (spotřeba 2,309 W - úspora 0,005 W) a kdy dochází k přenosu souborů do zařízení

Tab. 5.4: Tabulka naměřených hodnot po provedení optimalizace aktivních procesů a zápisů do interního úložiště zařízení.

Distribuce	Stav nečinnosti	Přenos souborů	100% využití procesoru
Rasperry Pi OS	2,309 W	2,541 W	3,523 W
Ubuntu	2,598 W	2,793 W	3,648 W
CentOS	2,445 W	2,556 W	3,280 W

(spotřeba 2,541 W - úspora 0,016 W). Distribuce CentOS je opět nejúspornější pro scénář se 100% vytížením procesoru (3,280 W - úspora 0,010 W). Nejméně úspornou distribucí pro všechny scénáře zůstává Ubuntu.

5.4 Zhodnocení docílených výsledků optimalizace a výběr nejúspornější Linuxové distribuce

Jednotlivé výsledky všech měření jsou v tabulce 5.4.1. Z této tabulky je patrné, že u všech distribucí došlo, na základě provedení výše popsanych úprav, ke značné úspoře spotřeby elektrické energie.

Nejvyšší úsporu můžeme pozorovat u distribuce CentOS, kdy pro 1. scénář došlo k úspoře 0,479 W, pro 2. scénář 0,477 W a pro 3. scénář k úspoře 2,081 W.

Naopak nejnižší úsporu spotřeby vykazuje distribuce Ubuntu s úsporou pro 1. scénář 0,116 W, pro 2. scénář pouze 0,036 W a pro 3. scénář úsporu 1,759 W.

Z naměřených hodnot můžeme dále pozorovat, že u všech distribucí došlo k největší úspoře pro scénář s největší zátěží zařízení, což je vzhledem k omezení maximální pracovní frekvence procesoru ve všech případech očekávaný výsledek.

Vzhledem ke skutečnosti, že zařízení má být případně nasazeno jako akumulátorově napájený server pro sběr dat z bezdrátových čidel, lze říci, že 3. scénář, tedy scénář s největší zátěží zařízení, pro který je nejúspornější distribuce CentOS, s výslednou mediánovou hodnotou spotřeby 3,280 W, není pro výsledný výběr tak důležitý jako jsou zbylé dva scénáře. Lze totiž předpokládat, že případný zápis dat ze senzorů a jejich následné odesílání bude zařízení, v porovnání se scénářem se 100% zatížením procesoru, zatěžovat velmi málo. Kvůli této skutečnosti tedy lze stanovit, že distribuce Raspberry Pi OS se jeví, jako nejúspornější distribuce, i když největší úspory pro všechny scénáře bylo dosaženo právě pro distribucí CentOS. Výsledné

Tab. 5.5: Tabulka porovnání hodnot po provedení změn v jednotlivých distribucích vedoucích ke snížení spotřeby zařízení Raspberry Pi 4 model B.

Distribuce	1. scénář	2. scénář	3. scénář
Rasperry Pi OS - výchozí nastavení	2,430 W	2,640 W	5,091 W
Rasperry Pi OS - základní úpravy	2,314 W	2,557 W	3,545 W
Rasperry Pi OS - úpravy procesů a zápisu do úložiště	2,309 W	2,541 W	3,523 W
Ubuntu - výchozí nastavení	2,714 W	2,829 W	5,407 W
Ubuntu - základní úpravy	2,601 W	2,819 W	3,666 W
Ubuntu - úpravy procesů a zápisu do úložiště	2,598 W	2,793 W	3,648 W
CentOS - výchozí nastavení	2,924 W	3,033 W	5,361 W
CentOS - základní úpravy	2,451 W	2,561 W	3,290 W
CentOS - úpravy procesů a zápisu do úložiště	2,445 W	2,556 W	3,280 W

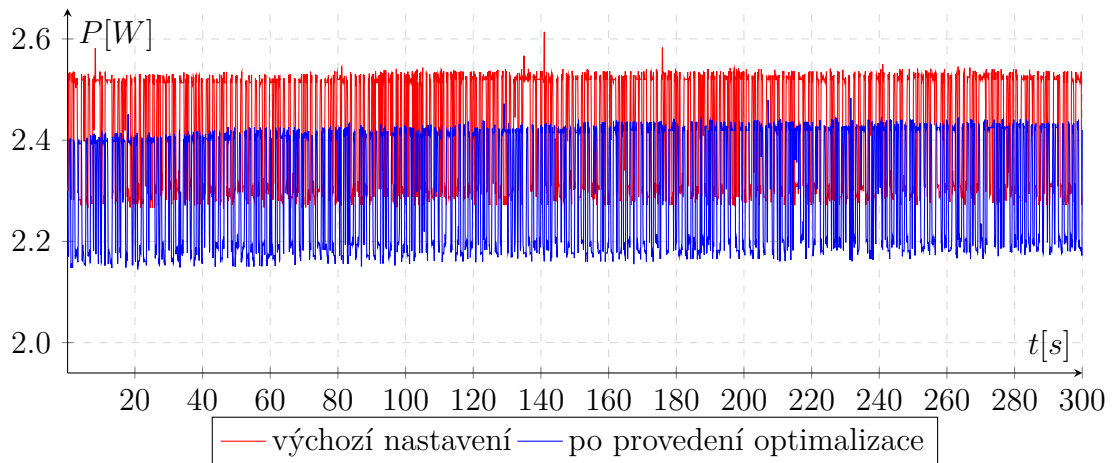
mediánové hodnoty spotřeby elektrické energie zařízení s distribucí Raspberry Pi OS jsou pro 1. scénář 2,309 W (celková úspora 0,121 W), pro 2. scénář 2,541 W (celková úspora 0,099 W) a pro 3. scénář 3,523 W (celková úspora 1,568 W).

5.4.1 Porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení optimalizace

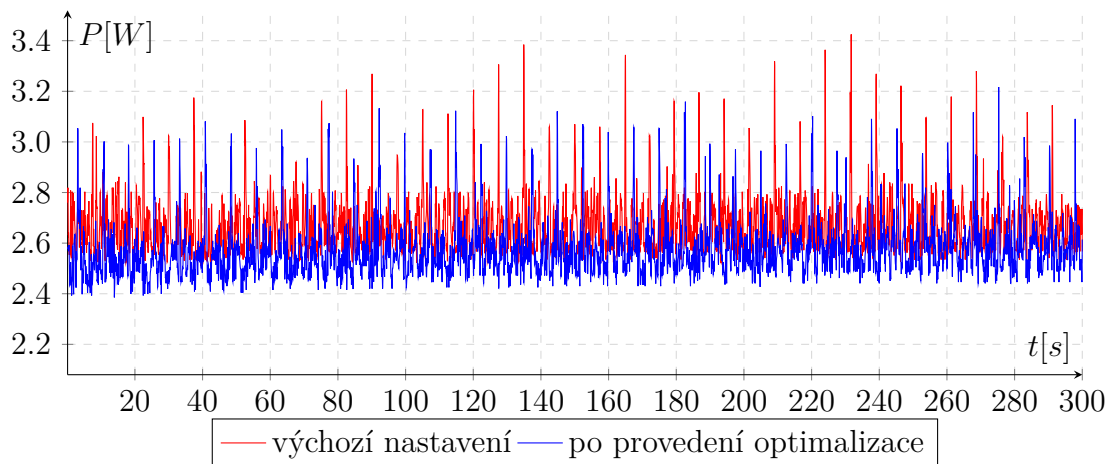
Nejúspornější distribucí pro zařízení Raspberry Pi 4, je Raspberry Pi OS. V této části je proto porovnána spotřeba zařízení pro všechny scénáře před a po provedení všech změn vedoucích k úspoře spotřeby elektrické energie.

Výsledné porovnání spotřeby zařízení s distribucí Raspberry Pi OS je zobrazeno na obrázcích 5.12, 5.13, 5.14 a v tabulce 5.6.

K nejmenší dosažené úspoře došlo pro scénář se střední zátěží. Jedním z faktorů proč v tomto případě nebylo dosaženo takové úspory, jako v ostatních případech je skutečnost, že na rozdíl od ostatních scénářů je výsledek méně ovlivněn rekonfigurací minimální a maximální pracovní frekvence procesoru, protože během přenosu souborů před i po úpravě minimální a maximální pracovní frekvence procesoru těchto hodnot nedosahuje.

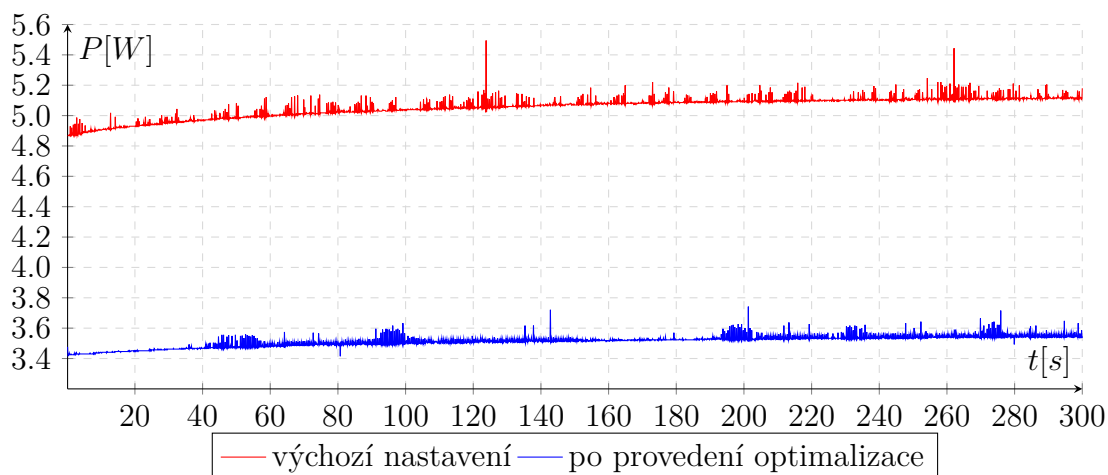


Obr. 5.12: Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář s minimální zátěží.



Obr. 5.13: Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář se střední zátěží.

Jak již bylo řečeno v předešlé části práce, k jednoznačně nejvyšší úspoře došlo pro scénář s maximální zátěží, což je patrné i z grafu na obrázku 5.14. Na tomto grafu si navíc můžeme všimnout, že po provedení všech úprav nejsou výkyvy ve spotřebě tak časté, jako je tomu ve výchozím nastavení a i nárůst spotřeby způsobený zvyšující se teplotou zařízení je pozvolnější.



Obr. 5.14: Graf porovnání spotřeby distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech změn pro scénář s maximální zátěží.

Tab. 5.6: Tabulka porovnání referenčních a finálních hodnot spotřeby pro Linuxovou distribuci Raspberry PI OS.

Distribuce	1. scénář	2. scénář	3. scénář
Raspberry Pi OS - výchozí nastavení	2,430 W	2,640 W	5,091 W
Raspberry Pi OS - po provedení všech úprav	2,309 W	2,541 W	3,523 W

Pro všechny scénáře došlo k určité úspoře spotřeby elektrické energie. Výsledné hodnoty úspory energie jsou pro scénář s nejmenším zatížením zařízení 0,121 W (úspora 4,979 %), pro scénář se střední zátěží 0,099 W (úspora 3,750 %) a pro scénář s největší zátěží zařízení 1,568 W (úspora 30,800 %). Procentuální úspory jsou pro první dva scénáře pouze v jednotkách procent, takže lze konstatovat, že zařízení je ve výchozím nastavení od výrobce velmi dobře optimalizované, pokud je zařízení v nečinnosti, nebo je zatěžováno do takové míry, která odpovídá předpokládané zátěži v normálním provozu zařízení. I když je měřená úspora pro scénář s největším zatížením 30,800 %, je nutno konstatovat, že po provedení změn, díky kterým došlo k této úspoře, zařízení nyní nedosahuje takového výpočetního výkonu, jako před provedením těchto změn, protože velká část této úspory je právě díky omezení maximální pracovní frekvence procesoru zařízení. Pro případné nasazení zařízení jako server pro sběr dat z bezdrátových čidel, je však i takto omezený výpočetní výkon dostačující.

6 Externí řízení napájení

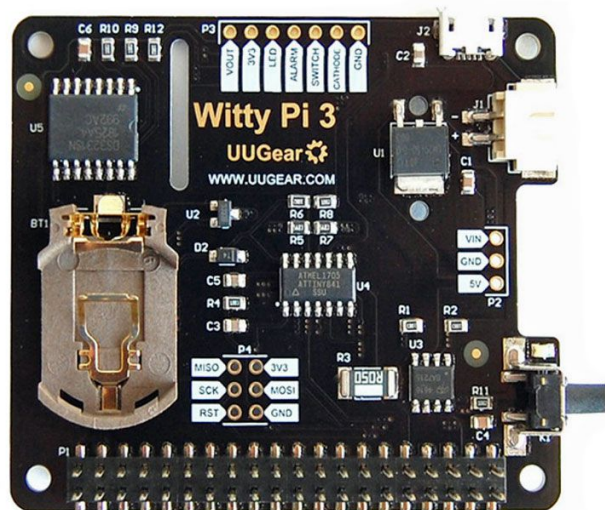
Vzhledem k zamýšlenému nasazení zařízení Raspberry Pi, jako serveru pro sběr dat z různých čidel, je možné docílit další úspory ve spotřebě vypínáním zařízení v době, kdy daná data není potřeba shromažďovat. Raspberry Pi 4B neobsahuje vnitřní hodiny (čas je získáván z připojení k internetu) a neexistuje zde tudíž možnost časovaného probuzení ze spánku.

Další skutečností je, že v případě, kdy je zařízení vypnuté (pomocí příkazu *halt*) je spotřeba zařízení stále velmi vysoká (1,750 W). V tomto stavu lze dosáhnout další úspory konfigurací bootladeru tak, aby ve stavu *halt* došlo k odpojení GPIO (general-purpose input/output) pinů od napájení. Po provedení tohoto nastavení je průměrná spotřeba zařízení ve stavu *halt* (tedy zařízení je vypnuté) 0,210 W. Díky tomuto nastavení však nadále není možné využít GPIO pinů pro krátkodobé odpojení napájení a tím znovu nastartování zařízení. Kvůli těmto nedostatkům zařízení Raspberry Pi je třeba využít externího zařízení pro řízení napájení.

6.1 Witty Pi 3

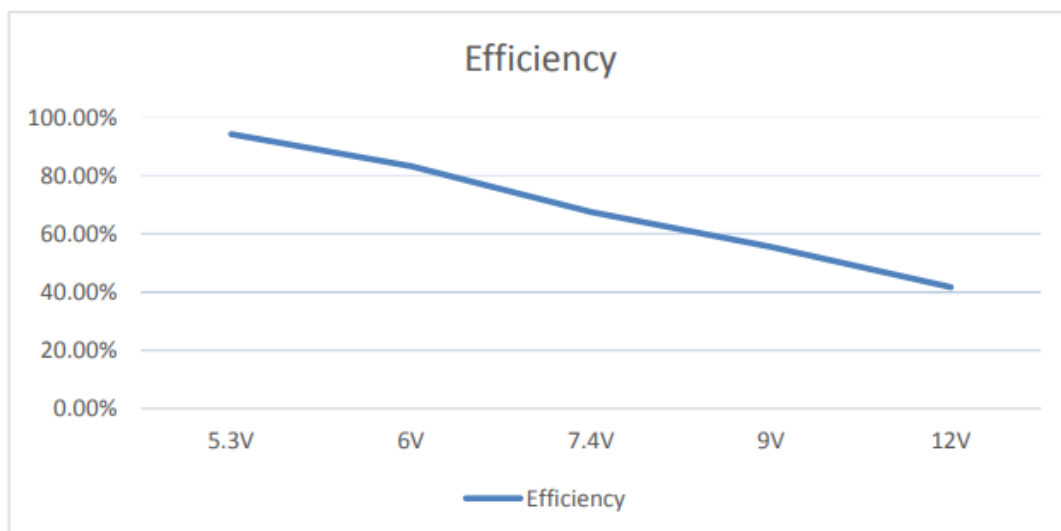
Modul Witty Pi 3 je HAT (Hardware attached on top) pro zařízení Raspberry Pi. Modul Witty Pi obsahuje obvod reálného času a je schopen plně spravovat napájení pro Raspberry Pi. Pomocí modul je možné nastavit plánované vypínání a zapínání zařízení Raspberry Pi ať už pomocí uživatelem vytvořených skriptů, nebo konzolového rozhraní s přesností na sekundy. Další výhodou je skutečnost, že před samotným odpojením napájení od Raspberry Pi je provedeno korektní vypnutí zařízení, čímž je zamezeno poškození dat, která může vzniknout okamžitým odpojením napájení ve chvíli, kdy je zařízení stále v provozu. Modul Witty Pi tedy odpojuje napájení Raspberry Pi až v momentě, kdy je zařízení ve vypnutém stavu.

Obvod reálného času modulu Witty Pi je v případě, kdy je zařízení Raspberry Pi vypnuté, napájeno z baterie typu CR2032. Díky tomu je spotřeba zařízení Raspberry Pi ve vypnutém stavu rovna 0 W. Výrobcem udávaná životnost této baterie pro modul Witty Pi je zhruba jeden rok [26]. V případě, že je zařízení napájeno z akumulátoru, který přechází do režimu spánku pokud nedetekuje po určité časové období žádné připojené zařízení, lze na modulu Witty Pi nastavit takzvaný *dummy load*, neboli fiktivní zátěž. Díky této funkci akumulátor do režimu spánku nepřechází. Když je funkce fiktivní zátěže aktivována, Witty bude odebírat v pevně daných krátkých časových intervalech pulzní proud (se špičkovou hodnotou 0,01 A), čímž je zabráněno přechodu akumulátoru do režimu spánku. V rámci diplomové práce však tato funkce není využita [26].



Obr. 6.1: Modul Witty Pi 3.

Modul Witty Pi 3 disponuje dvěma konektory pro napájení. Prvním z nich je konektor Micro-USB B a druhým XH2.54. Konektor XH2.54 je opatřen regulátorem napětí (LM29150), takže je možné zařízení napájet napětím, které je větší, než 5 V. Výrobce doporučuje maximální napájecí napětí je až 26 V. Protože LM29150 je lineární regulátor, poklesnuté napětí se rozptýlí v podobě tepla. Maximální výrobce doporučuje napětí, bez použití přídavného chladiče je tedy pouze 8 V [27]. Dalším efektem použití většího napájecího napájení je snížení celkové efektivity přenosu elektrické energie. Graf efektivity přenosu elektrické energie v závislosti na napájecím napětí je zobrazen na obrázku 6.2. Další výhodou při použití konektoru XH2.54 pro napájení je možnost nastavení minimálního prahového napětí a tím pádem zajištění stability zařízení Raspberry Pi (v případě napájecího napětí < 4 V dochází ke snížení stability zařízení, což se může projevit poškozením dat na paměťové kartě zařízení). Pro zajištění maximální úspory ve spotřebě je však v rámci práce využít pro napájení konektor typu Micro-USB B (5,25 V), který tyto funkce nepodporuje.



Obr. 6.2: Graf efektivitý přenosu elektrické energie v závislosti na napájecím napětí udávána výrobce modulu Witty Pi 3 (převzato z [26]).

V rámci diplomové práce bylo pro konfiguraci modulu Witty Pi použito terminálové uživatelské rozhraní, které je zobrazeno na ve výpisu 6.1. Pro účely diplomové práce bylo využito funkcí *Synchronize time* (pro synchronizaci času zařízení Raspberry Pi a modulu Witty Pi), *Schedule next shutdown* a *Schedule next startup* (pro nastavení plánovaných vypnutí a zapnutí zařízení Raspberry Pi). Jednotlivé funkce jsou číselně označeny a jsou volány zadáním příslušného čísla do terminálu.

Pro nastavení plánovaného spuštění a vypnutí zařízení je po zvolení příslušné funkce třeba zadat požadovaný čas ve správném formátu. Tento formát je ve tvaru DD HH:MM:SS. DD představuje den v měsíci, HH hodinu (dvaceti čtyř hodinový formát), MM minutu a SS vteřinu. Pokud je čas zařízení stejný, jako je definovaný čas, dojde k provedení dané akce. Pro opakovanou událost lze použít symbolu *?*. V případě že tento symbol použijeme například místo stanovení konkrétního dne a další údaje nastavíme na 13:00:00 bude k této akci docházet každý den přesně ve 13 hodin. Příklady různého nastavení s použitím symbolu *?* jsou ukázány v tabulce 6.1.

Výpis 6.1: Ukázka terminálového uživatelského rozhraní Witty Pi 3.

```

1 =====
2 |
3 |   Witty Pi - Realtime Clock + Power Management for Raspberry Pi   |
4 |
5 |               < Version 3.13 >           by UUGear s.r.o.           |
6 |
7 =====
8 >>> Current temperature: 27.0 C / 80.6 F
9 >>> Your system time is: Sun 11 Apr 2021 12:33:19 CET
10 >>> Your RTC time is:      Sun 11 Apr 2021 12:33:19 CET
11 >>> Vout=5.07V, Iout=0.51A
12 Now you can:
13   1. Write system time to RTC
14   2. Write RTC time to system
15   3. Synchronize time
16   4. Schedule next shutdown
17   5. Schedule next startup
18   6. Choose schedule script
19   7. Set low voltage threshold
20   8. Set recovery voltage threshold
21   9. View/change other settings...
22  10. Reset data...
23  11. Exit
24 What do you want to do? (1~11)

```

Tab. 6.1: Tabulka možných nastavení s použitím symbolu ? bez použití nastavení vteřin.

den	hodina	minuta	výsledek
??	??	??	akce se provede každou minutu (není doporučeno používat)
??	??	hodnota	akce se provede každou hodinu v zadanou minutu
??	hodnota	hodnota	akce se provede každý den v zadanou hodinu a minutu
hodnota	hodnota	hodnota	akce se provede pouze v zadaný den, hodinu a minutu

7 Simulace nasazení zařízení jako server pro koncentraci dat z různých čidel

V rámci simulace plní zařízení Raspberry Pi funkci serveru pro sběr dat z různých čidel. K simulaci byla vzhledem k výsledkům z předešlé části práce vybrána Linuxová distribuce Raspberry Pi OS. Přímě k zařízení je v tomto scénáři připojen modul měření teploty a tlaku BME280. Dále je využit modul ESP32-WROOM-32, ke kterému je připojen další modul pro měření teploty a vlhkosti vzduchu AOSONG DHT11. Tento modul představuje bezdrátové čidlo. Komunikace mezi ESP32 a Raspberry Pi probíhá skrze komunikační protokol MQTT (Message Queuing Telemetry Transport) prostřednictvím Wi-Fi sítě. Zařízení Raspberry Pi je v roli *Broker* (server pro sběr dat) a *Subscriber* (komponent pro přihlašování k odběru dat). ESP32 poté plní roli *Publisher* (generátor a odesílatel dat).

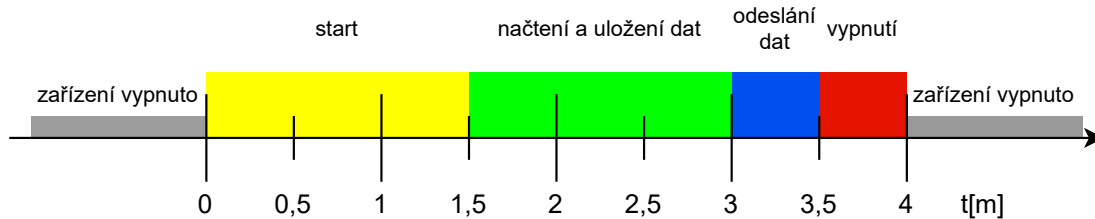
Jednotlivé veličiny jsou měřeny pro dva fiktivní objekty, kterými jsou chata a skleník. Raspberry Pi spolu s čidlem BME280 je umístěno v chatě, zatímco ESP32 s AOSONG DHT11 čidlem je umístěno ve skleníku. I když je pro komunikaci skrze protokol MQTT použito pouze jedno bezdrátové čidlo, jsou zde implementována MQTT témata *Cabin* a *Greenhouse*, pro snadné rozšíření dalšími bezdrátovými čidly. K odběru dat z případných dalších senzorů se lze jednoduše přihlásit podle již zmíněných témat.

V rámci simulace je k zařízení Raspberry Pi připojeno napájení pouze 4 minuty v každé hodině. Konkrétně je napájení připojeno každou celou hodinu (například 12:00) a odpojeno o 4 minuty později. Řízení zapínání a následného vypínání je plně řízeno modulem Witty Pi.

Vzhledem ke skutečnosti, že zařízení Raspberry Pi je v provozu vždy jen 4 minuty v každé hodině, není vhodné naměřená data na zařízení dále zpracovávat a zobrazovat například ve formě webové aplikace, protože by data byla k dispozici pouze v době, kdy je zařízení aktivní. Raspberry Pi proto naměřená data vždy ukládá do svého lokálního úložiště (uložené soubory jsou inkrementovány) a následně je odesílá na vzdálený SFTP server, kde mohou být dále zpracovávány.

Jednotlivé kroky, které zařízení periodicky každou hodinu v rámci simulace provádí jsou graficky znázorněny na obrázku 7.1.

Šedé bloky s označením *zařízení vypnuto* představují stav, ve kterém je zařízení Raspberry Pi odpojeno od napájení. Protože je modul Witty Pi v tomto režimu napájen vlastní baterií, výsledná spotřeba elektrické energie z externího zdroje (akumulátoru) je v tomto stavu 0 W.



Obr. 7.1: Grafické znázornění stavů zařízení Raspberry Pi v závislosti na čase.

V rámci bloku s označením *start* dochází k připojení napájení k zařízení Raspberry Pi, zavedení operačního systému, přihlášení k Wi-Fi síti a k MQTT komunikaci s ESP32. Jak již bylo zmíněno výše, připojení napájení k zařízení Raspberry Pi je spravováno modulem Witty Pi. Přihlašování k Wi-Fi síti i k MQTT komunikaci je řízeno autonomně pomocí skriptů, které spouští softwarový démon *cron*.

V dalším bloku označeném *načtení a uložení dat* dochází k načtení dat z čidel a jejich následné uložení do souborů ve formátu CSV. Data jsou do těchto souborů postupně inkrementována. Očekávána velikost těchto souborů po jednom měsíci zaznamenávání hodnot je 32,3 kB. I když je tato očekávaná velikost relativně malá, není vhodné neustále přidávat nové hodnoty do stejného souboru. Je proto vhodné zajistit, aby byl každý měsíc generovaný a odesílaný nový soubor a hodnoty z daného měsíce zapisovány pouze do něj. V rámci diplomové práce však tento mechanismus není zaveden. Protože bylo měření prováděno v zarušeném prostředí (během psaní diplomové práce nebylo možné využít školní laboratoře kvůli koronavirové krizi) často docházelo ke ztrátám dat v rámci MQTT komunikace. Je zde proto použito QOS 2 (Quality of Service), tedy exactly once (právě jednou). Tím je zaručeno, že zpráva vždy dojde právě jednou, ale zároveň dochází ke značnému zpomalení celé komunikace. V nezarušeném prostředí by tento blok mohl být mnohem kratší, což by mělo za následek snížení celkové spotřeby elektrické energie.

V bloku *odeslání dat* dochází k připojení ke vzdálenému SFTP a následnému odeslání CSV souborů s naměřenými daty na tento server.

V posledním bloku s označením *vypnutí* modul Witty Pi odešle signál pro vypnutí na zařízení Raspberry Pi, které zahájí proces vypnutí. Jakmile je Raspberry Pi zcela vypnuté, modul Witty Pi provede kompletní odpojení zařízení Raspberry Pi od napájení.

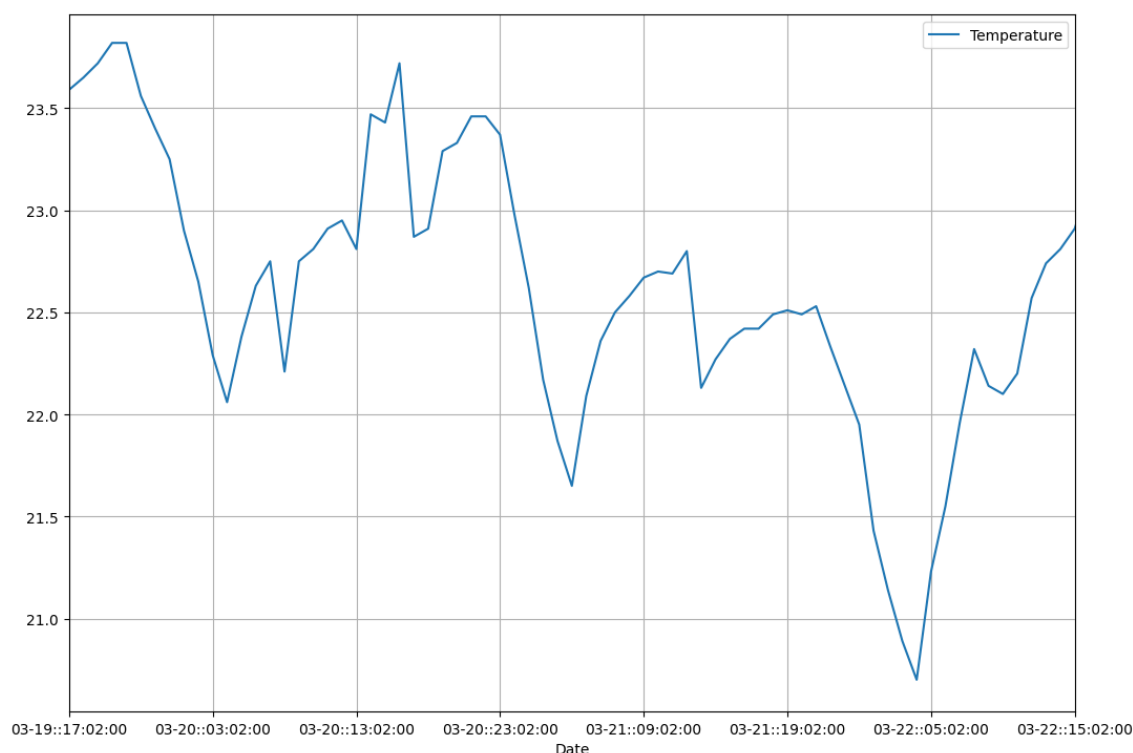
7.1 Zpracování naměřených dat z čidel

Jak již bylo zmíněno v předešlé části práce, naměřené hodnoty není vhodné dále zpracovávat na zařízení Raspberry Pi, kvůli skutečnosti, že je toto zařízení v zapnutém stavu pouze 4 minuty v každé hodině (dosažitelné například pomocí SSH pouze 2 minuty).

K dalšímu zpracování naměřených hodnot a jejich případné vynášení do grafů je využíván vzdálený SFTP server. Protože jsou jednotlivé soubory před odesláním z Raspberry Pi inkrementovány o nové hodnoty nedochází ke ztrátě dat například pokud je SFTP server v moment odesílání dat nedostupný. V takovém případě dochází pouze ke zpoždění doručení těchto dat, protože Raspberry Pi vždy odesílá soubor s kompletními daty pro dané časové období.

Naměřené data lze následně zobrazovat například pomocí webové aplikace a mít k nim tak přístup odkudkoli.

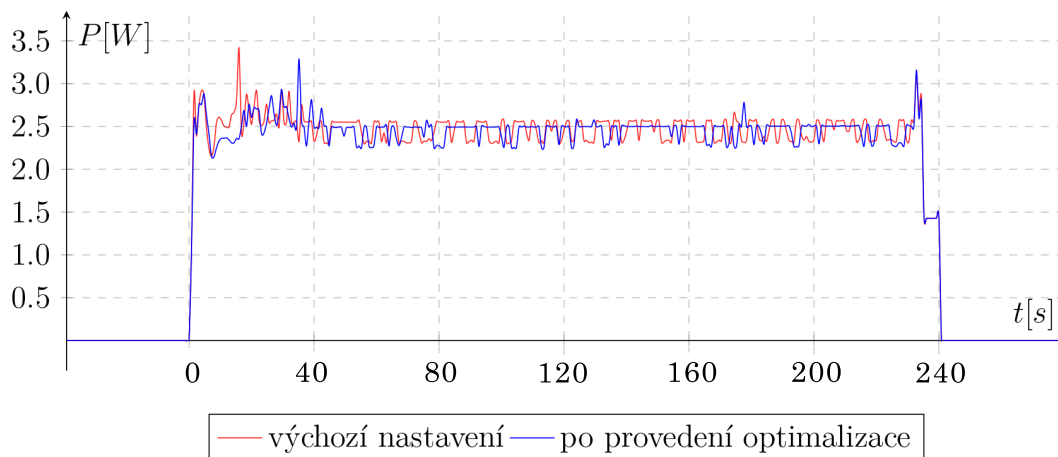
Ukázka možného zobrazení naměřených hodnot pomocí programovacího jazyka *python* a knihovny *Pandas* je na obrázku 7.2.



Obr. 7.2: Graf teploty ve fiktivním objektu *chata*.

7.2 Měření spotřeby zařízení v rámci testovacího scénáře

Simulace probíhala 72 hodin pro zařízení ve výchozím nastavení i po provedení optimalizačních změn. Na obrázku 7.3 je zachycena spotřeba zařízení jednoho cyklu, kdy je k zařízení připojeno napájení (od $t = 0$ s až po $t = 240$ s).

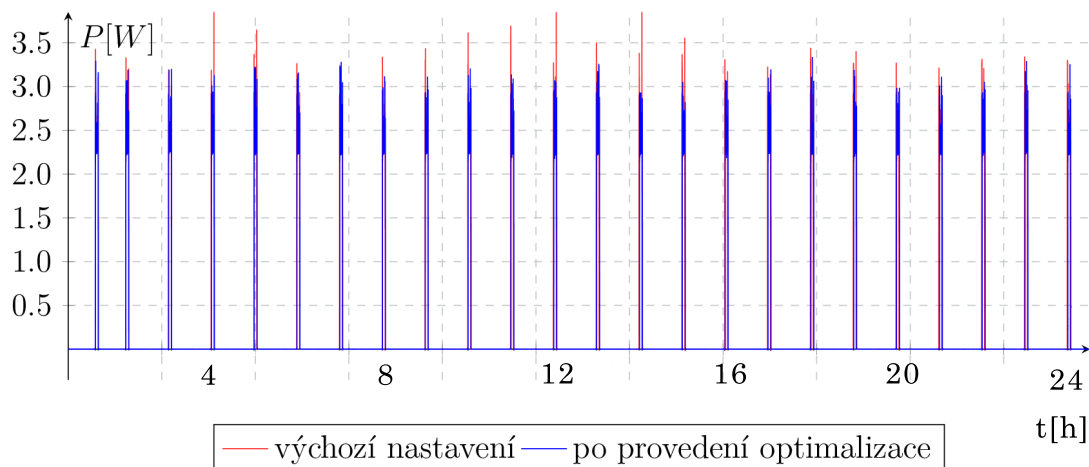


Obr. 7.3: Graf porovnání spotřeby zařízení v rámci jednoho cyklu.

Zařízení je plně schopno fungovat až zhruba v čase $t = 90$ s i když z grafu spotřeby tato změna stavu ze zavádění systému do plného provozu není pozorovatelná. Pozorovatelné však je připojení k SFTP serveru a následné odeslání naměřených dat zhruba v čase $t = 180$ s. V grafu si dále můžeme všimnout, že v případě zavádění operačního systému došlo ke značné úspoře spotřeby elektrické energie v případě použití optimalizačních nastavení. Například špičková hodnota spotřeby elektrické energie při zavádění činila u zařízení ve výchozím nastavení 3,420 W, zatímco po optimalizaci pouze 3,289 W. Další zajímavostí je fakt, že spotřeba v rámci vypínání zařízení je téměř totožná a optimalizační úpravy na tuto část grafu nemají žádný vliv.

Na grafu 7.4 je dále zobrazen graf spotřeby elektrické energie pro delší časový úsek. Pro přehlednost zde není vyneseno celých 72 hodin měření ale pouze 24 hodin. Na tomto grafu jsou více patrné špičkové rozdíly spotřeby elektrické energie při zavádění operačního systému, kdy maximální hodnota spotřeby pro zařízení ve výchozím nastavení činila 4,154 W, zatímco pro zařízení s implementovanými optimalizačními nastaveními pouze 3,336 W. Z tohoto grafu se může zdát, že výsledná úspora ve spotřebě elektrické energie je značná, ale ve skutečnosti se jedná o již výše

zmíněné špičkové hodnoty, které neovlivňují celkovou spotřebu takovou mírou, jako spotřeba v ostatních fázích chodu zařízení.



Obr. 7.4: Graf porovnání spotřeby zařízení během 24 hodin.

Výsledné hodnoty spotřebované elektrické energie během 72 hodin jsou uvedeny v tabulce 7.1.

Tab. 7.1: Porovnání spotřeby zařízení v rámci testovacího scénáře ve výchozím stavu a po provedení optimalizace (měřené napětí 5,25 V).

stav	Spotřeba zařízení za 72 hodin
výchozí nastavení	11,7822 Wh (2,2442 Ah)
po provedení optimalizace	11,6343 Wh (2,2161 Ah)

Z hodnot z tabulky 7.1 lze určit, že výsledná úspora je asi 1,255 %. Tato hodnota je velmi odlišná od očekávané hodnoty úspory 3,750 % z kapitoly 5.4 (druhý scénář). Tato odlišnost je způsobena cyklickým zapínáním a vypínáním zařízení. Stavy zavádění systému, ani vypínání zařízení, nebyly z hlediska spotřeby popsány a nejsou proto zahrnuty do výpočtů spotřeby v předešlých kapitolách.

Pro jiný scénář, ve kterém by zařízení bylo po delší časové úseky v zapnutém stavu a docházelo by k menšímu počtu odpojení a připojení napájení k zařízení lze očekávat, že by se výsledná hodnota úspory elektrické energie více přiblížila k očekávané hodnotě úspory 3,750 %.

8 Akumulátorové napájení

Pokud budeme uvažovat fiktivní akumulátor typu Li-Pol (Lithium-polymer) a požadujeme funkčnost zařízení Raspberry Pi při napájení z tohoto akumulátoru alespoň měsíc, můžeme určit jeho požadovanou kapacitu z měření v předešlé části práce na 22,161 Ah. Dále je ale třeba uvažovat samovolné vybíjení akumulátoru, které u tohoto typu akumulátoru činí asi 5 % jeho kapacity za měsíc [28]. Dále není vhodné akumulátor vybíjet úplně, ale využívat pouze 80 % jeho kapacity, což vede k prodloužení životnosti až o 50 % cyklů [29]. V takovém případě je potřebná kapacita akumulátoru 27,701 Ah. Pokud budeme uvažovat i určitou rezervu, lze stanovit výslednou kapacitu akumulátoru na 30 Ah, z kterých máme k dispozici 22,5 Ah pro napájení zařízení.

Pomocí naměřených hodnot spotřeby z předešlé kapitoly můžeme dále určit, jak dlouho bude zařízení schopné fungovat při napájení z akumulátoru s kapacitou 30 Ah. Výsledné hodnoty jsou uvedeny v tabulce 8.1.

Tab. 8.1: Doba provozu zařízení při napájení z akumulátoru.

stav	Doba provozu
výchozí nastavení	721,9 hodin
po provedení optimalizace	731,0 hodin

Zařízení je v obou případech schopno fungovat při akumulátorovém napájení více, než 30 dní. V případě zařízení ve výchozím nastavení se jedná o dobu 30 dní a jedné celé hodiny. V případě, kdy jsou na zařízení provedeny optimalizační úpravy se jedná o dobu 30 dní a 11 hodin. Implementováním optimalizačních úprav se tedy doba provozu prodloužila o 10 hodin. Jedná se o relativně malou úsporu, což je ale očekávaný výsledek, vzhledem k výsledkům z předešlé části práce.

9 Ovlivnění funkčnosti zařízení optimalizačními úpravami

Optimalizační úpravy provedené v rámci práce ovlivňují funkčnost samotného zařízení. Mezi největší omezení patří nemožnost využití USB portů zařízení, díky čemuž není například možné připojit k zařízení externí úložiště pro ukládání dat ze senzorů. Dalším omezením je úplné vypnutí Bluetooth, což zamezuje veškeré komunikaci skrze toto rozhraní (nelze například používat čidla komunikující skrze Bluetooth rozhraní).

Dalším velkým zásahem do funkčnosti zařízení je omezení maximálního výpočetního výkonu zařízení. Výpočetní schopnosti zařízení byly zredukovány zhruba o 60 % oproti výchozímu nastavení. V rámci diplomové práce sice nebyla pozorována žádná změna v chování zařízení, ale v případě většího zatížení zařízení je výkon podstatně zredukován.

Velkou nevýhodou takto upraveného zařízení je i absence záznamů systémových událostí (logů). Toto omezení může být problematické například při využívání většího množství bezdrátových čidel, kdy může být složitější bez těchto záznamů dohledat zdroje případných chyb spojení.

K zařízení není možné po provedení optimalizačních úprav připojit myš, klávesnici ani monitor. Vzhledem k definovanému nasazení zařízení však tyto periferie nejsou využívány. Jediným způsobem jak zařízení ovládat a spravovat je přístup skrze Wi-Fi rozhraní (pomocí SSH protokolu) a to pouze v omezených časových intervalech (první 4 minuty v každé hodině). Velkou nevýhodou však představuje nemožnost správy zařízení v případě změn v síti (případně úplnému kolapsu Wi-Fi sítě). Pokud by k takové změně došlo zařízení nelze nijak ovládat a přenastavit.

10 Shrnutí dosažených výsledků

Raspberry Pi 4 je vzhledem ke své ceně a velikosti velmi výkonným a všestranným zařízením, jehož spotřeba je již ve výchozím nastavení relativně malá. V porovnání s jinými zařízeními běžně používanými v rámci IOT (Internet of Things) projektů, u nichž je běžné nasazování s akumulátorového napájení, je však spotřeba zařízení Raspberry Pi mnohem větší. Například zařízení Arduino Uno ve výchozím nastavení vykazuje průměrnou spotřebu 885,87 mW [30], zatímco spotřeba Raspberry Pi 4 je ve výchozím nastavení 2,43 W, tedy asi 2,75 krát více. I když zařízení Raspberry Pi 4 vykazuje podstatně vyšší spotřebu, než v příkladu uvedené Arduino Uno, poskytuje oproti ostatním řešením velkou výhodu díky svému výkonu a mnohem větší všestrannosti (i díky možnosti instalace různých Linuxových distribucí).

V rámci diplomové práce bylo dosaženo určité úspory (přesné hodnoty jsou uvedeny v kapitole 5.4.1) v rámci jednotlivých scénářů, které byly definovány v kapitole 5. Této úspory bylo dosaženo díky různým optimalizačním změnám, které mají za následek snížení spotřeby elektrické energie, ale i do jisté míry omezení schopností zařízení Raspberry Pi 4 (omezení těchto schopností je popsáno v kapitole 9).

Optimalizační úpravy byly dále testovány v rámci simulace reálného nasazení Raspberry Pi 4 jako serveru pro sběr dat z různých čidel. V rámci tohoto scénáře byla hodnota spotřebované elektrické energie značně odlišná, než byla její očekávaná hodnota, vzhledem k předchozím měřením (přesné hodnoty a odůvodnění odlišnosti jsou uvedeny v kapitole 7.2).

Při použití fiktivní baterie s kapacitou 30 Ah došlo díky optimalizačním úpravám k prodloužení životnosti takto napájeného zařízení Raspberry Pi 4, v rámci simulace reálného nasazení o 10 hodin provozu (v původním stavu 721 hodin, po provedení optimalizace 731 hodin). Z měření tedy vyplývá, že spotřeba elektrické energie zařízení Raspberry Pi 4 je i po provedení v práci popsaných optimalizačních úprav poměrně vysoká (úspora pouze 1,255 %) a tudíž se nejedná o zařízení vhodné k realizaci IOT projektů, u kterých je třeba využití akumulátorově napájeného zařízení.

Závěr

Tato diplomová práce se zabývá optimalizací různých Linuxových distribucí dostupných pro zařízení Raspberry Pi 4 model B a následným výběrem nejúspornější z nich. Optimalizace je prováděna s ohledem na zamýšlené využití zařízení Raspberry Pi jako serveru pro sběr dat z různých čidel. Zvolené Linuxové distribuce jsou Raspberry Pi OS, Ubuntu a CentOS.

V práci bylo popsáno samotné zařízení Raspberry Pi 4 model B, vybrané distribuce a použité softwarové nástroje, které byly použity pro snížení spotřeby zařízení.

V první praktické (optimalizační) části práce byly popsány jednotlivé scénáře měření spotřeby. Tyto scénáře představovaly různé stupně zatížení zařízení. První scénář byl zaměřen na spotřebu zařízení v klidovém režimu, tedy když je zařízení v nečinnosti. Druhý scénář představoval režim se středním zatížením. V rámci toho scénáře docházelo k přenosu souborů do zařízení pomocí protokolu SFTP s omezenou rychlostí na 1 MB/s. V rámci posledního scénáře bylo zařízení (konkrétně procesor) zatěžováno na 100 % výpočetního výkonu.

V rámci výběru nejúspornější Linuxové distribuce proběhlo nejprve měření pro všechny distribuce ve výchozím nastavení. Tímto měřením byly získány referenční hodnoty spotřeby pro další porovnání spotřeby po provedení optimalizačních úprav. Práce se dále zabývá popisem a provedením základních úprav pro snížení spotřeby, orientovaných spíše na hardware zařízení. Po provedení změn proběhlo další měření a následné srovnání spotřeby jednotlivých distribucí. Další část práce se zabývala optimalizací softwarových procesů a aplikací. Jednalo se zejména o odstranění, nebo vypnutí nepotřebných služeb a kontroly zdali je zařízení ve stavu nečinnosti skutečně zcela nečinné. V rámci všech distribucí došlo k uvedení zařízení do stavu, kdy dochází k minimální zátěži zařízení běžícími procesy. Jediné procesy, které zůstaly aktivní jsou takové, které jsou nezbytně nutné pro správné fungování zařízení. Po provedení těchto úprav následovalo měření, srovnání výsledků a výběr nejúspornější distribuce, kterou bylo Raspberry Pi OS. V práci jsou dále srovnány referenční hodnoty spotřeby elektrické energie (tedy hodnoty spotřeby ve výchozím nastavení) a hodnoty spotřeby po provedení všech v práci popsaných úprav pro distribuci Raspberry Pi OS.

Práce se dále zabývala možným řešením externího napájení zařízení, které vede k další možné úspoře v případě reálného nasazení zařízení. Externí napájení Zařízení Raspberry Pi bylo v práci řešeno pomocí zařízení Witty Pi 3. Následně došlo k navržení simulačního scénáře reálného nasazení zařízení a měření hodnot spotřeby

zařízení při použití distribuce Raspberry Pi OS ve výchozím nastavení a po provedení všech optimalizačních úprav.

Z naměřených hodnot spotřeby v rámci scénáře simulujícího reálné nasazení zařízení bylo následně stanoveno, jaký dopad mají optimalizační úpravy na výslednou výdrž zařízení při použití akumulátorového napájení. Při použití 30 Ah akumulátoru tedy došlo o prodloužení životnosti o 10 hodin (z původních 721 hodin na 731 hodin nepřetržitého provozu). V rámci této simulace tedy bylo dosaženo úspory spotřeby elektrické energie o 1,255 %.

Z naměřených výsledků lze říci, že zařízení Raspberry Pi 4 je vynikající zařízení v pro použití v různých IoT projektech (vzhledem ke svému výkonu a všestrannosti). Pro projekt, kde je však potřeba, aby zařízení bylo schopné fungovat po dlouhé časové úseky s napájením pouze z akumulátoru, je z hlediska spotřeby mnohem výhodnější použití jiného, méně všestranného zařízení (například Arduino Uno). Protože i přes použití optimalizačních úprav je výsledná spotřeba zařízení oproti jiným zařízením velmi velká.

Literatura

- [1] *Raspberry Pi 4. Raspberrypi [online]. UK: The Raspberry Pi Foundation [cit. 2020-10-14]* [online]. Dostupné z URL: <<https://www.raspberrypi.org/products/raspberry-pi-4-model-b/?resellerType=home>>.
- [2] *Raspberrypi.org [online]. UK: The Raspberry Pi Foundation [cit. 2020-10-14]* [online]. Dostupné z URL: <<https://www.raspberrypi.org/>>.
- [3] *Ubuntu.cz [online]. Canonical [cit. 2020-10-15]* [online]. Dostupné z URL: <<https://www.ubuntu.cz/o-ubuntu/>>.
- [4] *Ubuntu.com [online]. Canonical [cit. 2020-10-15]* [online]. Dostupné z URL: <<https://ubuntu.com/>>.
- [5] *CentOS.org [online]. CentOS Project [cit. 2020-10-17]* [online]. Dostupné z URL: <<https://www.centos.org/about/>>.
- [6] *Scribd.com [online]. 2020 Scribd Inc. [cit. 2020-11-03]* [online]. Dostupné z URL: <https://cz.gearbest.com/other-instruments/pp_009175502086.html?wid=1433363#goodsDetail>.
- [7] *Package: s-tui. Debian [online]. SPI Inc [cit. 2020-11-03]* [online]. Dostupné z URL: <<https://packages.debian.org/sid/utils/s-tui>>.
- [8] *top(1) - Linux man page. die.net [online]. die.net, 1996 [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://linux.die.net/man/1/top>>.
- [9] *powertop(8) - Linux man page. die.net [online]. die.net, 1996 [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://linux.die.net/man/8/powertop>>.
- [10] *perf(1) — Linux manual page. man7.org [online]. Michael Kerrisk [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://man7.org/linux/man-pages/man1/perf.1.html>>.
- [11] *iotop(1) - Linux man page. die.net [online]. die.net, 1996 [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://linux.die.net/man/1/iotop>>.
- [12] *Kernel-per-CPU-threads. Kernel.org [online]. California: Linux Kernel Organization [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://www.kernel.org/doc/Documentation/kernel-per-CPU-kthreads.txt>>.
- [13] *Kernel-per-CPU-threads. Mjmwired [online]. Mauriat Miranda, 2018 [cit. 2020-11-12]* [online]. Dostupné z URL: <<https://www.mjmwired.net/kernel/Documentation/kernel-per-CPU-kthreads.txt>>.

- [14] *Sshd. Ssh.com [online]. Communications Security [cit. 2020-11-12] [online].* Dostupné z URL: <<https://www.ssh.com/ssh/sshd/>>.
- [15] *Irqbalance. die.net [online]. die.net, 1996 [cit. 2020-11-12] [online].* Dostupné z URL: <<https://linux.die.net/man/1/irqbalance>>.
- [16] *Understanding and administering systemd. Fedora DOCS [online]. fedora DOCS [cit. 2020-11-12] [online].* Dostupné z URL: <<https://docs.fedoraproject.org/en-US/quick-docs/understanding-and-administering-systemd/>>.
- [17] *Systemd-udev.service. Ubuntu manuals [online]. California: Canonical [cit. 2020-11-12] [online].* Dostupné z URL: <<https://manpages.ubuntu.com/manpages/bionic/man8/systemd-udev.service.8.html>>.
- [18] *SYSTEMD-JOURNALD.SERVICE. Manapges debian [online].SPI, 2020 [cit. 2020-11-12] [online].* Dostupné z URL: <<https://manpages.debian.org/testing/systemd/systemd-journald.service.8.en.html>>.
- [19] *Avahi. Avahi [online].SPI, 2020 [cit. 2020-11-12] [online].* Dostupné z URL: <<https://www.avahi.org/>>.
- [20] *Thd [online].California: Canonical [cit. 2020-11-12]] [online].* Dostupné z URL: <<https://manpages.ubuntu.com/manpages/bionic/man1/thd.1.html>>.
- [21] *What is PackageKit? Freedesktop.org [online].Software in the Public Interest [cit. 2020-11-12] [online].* Dostupné z URL: <<https://www.freedesktop.org/software/PackageKit/pk-intro.html>>.
- [22] *A Beginners Guide to Snaps in Linux. TecMint [online].Tecmint [cit. 2020-11-12] [online].* Dostupné z URL: <<https://www.tecmint.com/install-snap-in-linux/>>.
- [23] *The Tuned Project. The Tuned Project[online].Fedora Project [cit. 2020-11-12] [online].* Dostupné z URL: <<https://tuned-project.org/>>.
- [24] *chronyd(8) - Linux man page. die.net [online]. die.net, 1996 [cit. 2020-11-12] [online].* Dostupné z URL: <<https://linux.die.net/man/8/chronyd>>.
- [25] *cauditd(8) - Linux man page. die.net [online]. die.net, 1996 [cit. 2020-11-12] [online].* Dostupné z URL: <<https://linux.die.net/man/8/auditd>>.
- [26] *Witty Pi 3 User Manual. UUGear [online]. Praha, Bělehradská 858/23: UU-Gear, 2020 [cit. 2021-04-11] [online].* Dostupné z URL: <http://www.uugear.com/doc/WittyPi3_UserManual.pdf>.

- [27] *LM29150 [online]. Praha, Bělehradská 858/23: UUGear, 2020 [cit. 2021-04-11] [online]. Dostupné z URL: <<http://www.uugear.com/doc/datasheet/LM29150.pdf>>.*
- [28] *ALAMEH, Ziyad a Bong Gon Kim. Advanced lithium polymer batteries. IEEE Xplore [online]. 1 [cit. 2021-04-09] [online]. Dostupné z doi: doi:10.1109/PES.2009.5275404.*
- [29] *BU-808: How to Prolong Lithium-based Batteries. Battery University [online]. Nürnberg, Německo: Cadex Electronics, 21.10. 2010n. l. [cit. 2021-04-09] [online]. Dostupné z URL: <https://batteryuniversity.com/learn/article/how_to_prolong_lithium_based_batteries>.*
- [30] *DAVID, Christopher. Arduino Uno Tutorial [Pinout]. DiyIoT [online]. DIYIoT, 2020 [cit. 2021-04-14] [online]. Dostupné z URL: <<https://diyi0t.com/arduino-uno-tutorial/>>.*

Seznam symbolů, veličin a zkratek

APT	Advanced Package Tool
ARM	Advanced RISC Machine
CentOS	Community Enterprise Operating System
CSV	Comma-separated values
DEB	Debian
GPIO	general-purpose input/output
HAT	Hardware attached on top
HDMI	High-Definition Multimedia Interface
IOT	Internet of Things
JBD2	Journaling Block Device version 2
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
Li-Pol	Lithium-polymer
LTS	dlouhodobá podpora – Long-Term Support
LZMA	Lempel-Ziv-Markov-Chain Algorithm
MIPS	million instructions per second
MQTT	Message Queuing Telemetry Transport
OS	Operating System
QOS	Quality of Service
RAM	Random Access Memory
RHEL	Red Hat Enterprise Linux
SFTP	Secure Shell File Transfer Protocol
SSH	Secure Shell
s-tui	Stress-Terminal User Interface

USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WPA2	Wi-Fi Protected Access version 2
Yum	Yellowdog Updater, Modified

Seznam příloh

A Obsah elektronické přílohy

60

A Obsah elektronické přílohy

/.....	kořenový adresář přiloženého archivu
Optimalizace.....	Naměřené hodnoty spotřeby v rámci optimalizace
CentOS	
Minimální činnost zařízení	
CentOS-minimální-scénář-1.csv	
CentOS-minimální-scénář-2.csv	
CentOS-minimální-scénář-3.csv	
Výchozí nastavení	
CentOS-výchozí-scénář-1.csv	
CentOS-výchozí-scénář-2.csv	
CentOS-výchozí-scénář-3.csv	
Základní úpravy	
CentOS-základní-scénář-1.csv	
CentOS-základní-scénář-2.csv	
CentOS-základní-scénář-3.csv	
Raspberry Pi OS	
Minimální činnost zařízení	
RpiOS-minimální-scénář-1.csv	
RpiOS-minimální-scénář-2.csv	
RpiOS-minimální-scénář-3.csv	
Výchozí nastavení	
RpiOS-výchozí-scénář-1.csv	
RpiOS-výchozí-scénář-2.csv	
RpiOS-výchozí-scénář-3.csv	
Základní úpravy	
RpiOS-základní-scénář-1.csv	
RpiOS-základní-scénář-2.csv	
RpiOS-základní-scénář-3.csv	
Ubuntu	
Minimální činnost zařízení	
Ubuntu-minimální-scénář-1.csv	
Ubuntu-minimální-scénář-2.csv	
Ubuntu-minimální-scénář-3.csv	
Výchozí nastavení	
Ubuntu-výchozí-scénář-1.csv	
Ubuntu-výchozí-scénář-2.csv	
Ubuntu-výchozí-scénář-3.csv	
Základní úpravy	
Ubuntu-základní-scénář-1.csv	
Ubuntu-základní-scénář-2.csv	
Ubuntu-základní-scénář-3.csv	
Simulace nasazení zařízení	Naměřené hodnoty spotřeby v rámci simulace
72 hodin provozu	
72-optimalizace.csv	

- └─ 72-původní.csv
- └─ Zavádění operačního systému
 - └─ Zavádění-optimalizace.csv
 - └─ Zavádění-původní.csv